

(19) 世界知的所有権機関  
国際事務局



(43) 国際公開日  
2003 年 12 月 24 日 (24.12.2003)

PCT

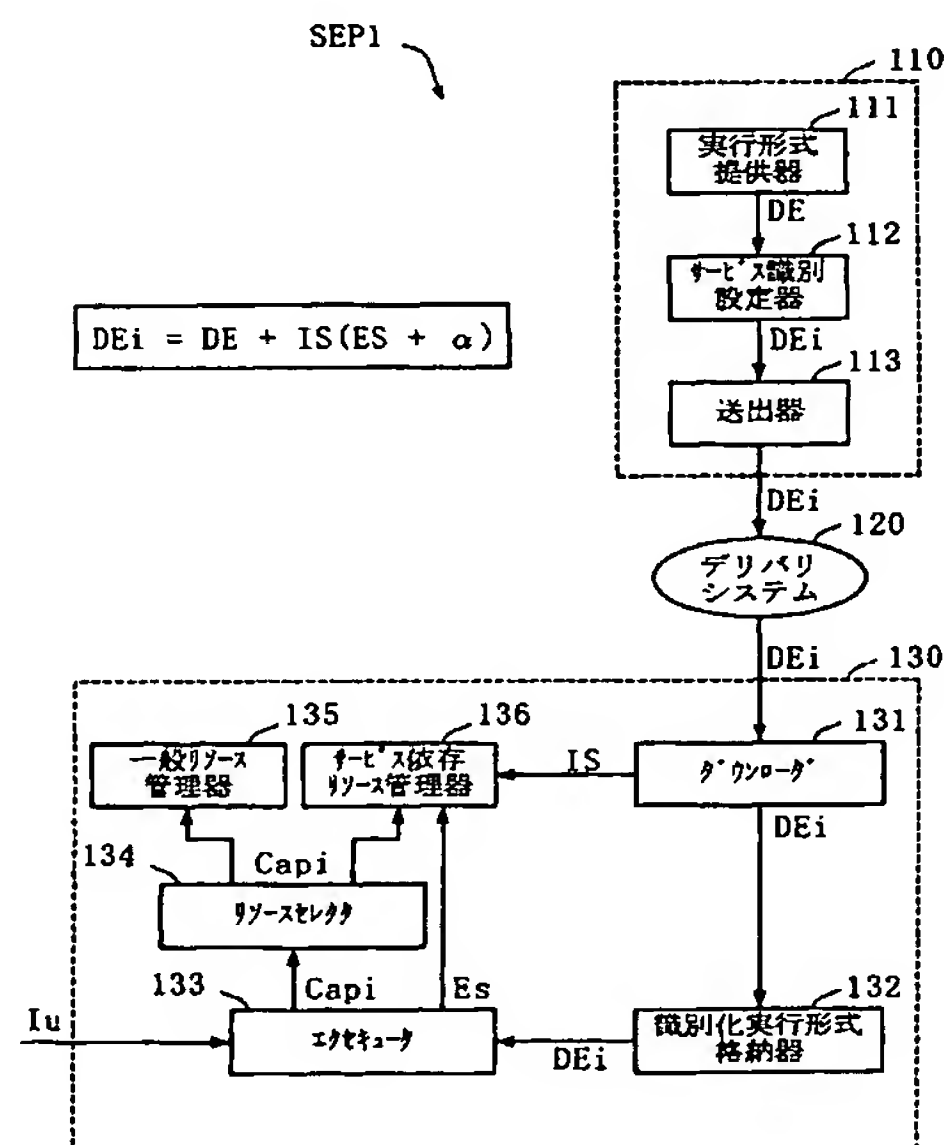
(10) 国際公開番号  
WO 03/107182 A1

- (51) 国際特許分類: G06F 9/445
- (21) 国際出願番号: PCT/JP03/07390
- (22) 国際出願日: 2003 年 6 月 11 日 (11.06.2003)
- (25) 国際出願の言語: 日本語
- (26) 国際公開の言語: 日本語
- (30) 優先権データ:  
特願 2002-171338 2002 年 6 月 12 日 (12.06.2002) JP
- (71) 出願人 (米国を除く全ての指定国について): 松下電器産業株式会社 (MATSUSHITA ELECTRIC INDUSTRIAL CO., LTD.) [JP/JP]; 〒571-8501 大阪府門真市大字門真 1006 Osaka (JP).
- (72) 発明者; および
- (75) 発明者/出願人 (米国についてのみ): 片岡 充照 (KATAOKA, Mitsuteru) [JP/JP]; 〒576-0034 大阪府交野市天野が原町 4-1 5-1 1 Osaka (JP).
- (74) 代理人: 小笠原 史朗 (OGASAWARA, Shiro); 〒564-0053 大阪府吹田市江の木町 3 番 1 1 号 第 3 ロンチェビル Osaka (JP).
- (81) 指定国 (国内): CN, JP, US.
- (84) 指定国 (広域): ヨーロッパ特許 (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HU, IE, IT, LU, MC, NL, PT, RO, SE, SI, SK, TR).
- 添付公開書類:  
— 国際調査報告書

[続葉有]

(54) Title: SERVICE SAFELY-EXTENSIBLE PLATFORM

(54) 発明の名称: サービス安全拡張プラットフォーム



- 111...EXECUTE FORM PROVIDING APPARATUS  
112...SERVICE IDENTIFICATION SETTING APPARATUS  
113...TRANSMISSION APPARATUS  
120...DELIVERY SYSTEM  
135...GENERAL RESOURCE MANAGEMENT APPARATUS  
136...SERVICE-DEPENDENT RESOURCE MANAGEMENT APPARATUS  
131...DOWNLOADER  
134...RESOURCE SELECTOR  
133...EXECUTOR  
132...IDENTIFIABLE EXECUTE FORM STORAGE APPARATUS

(57) Abstract: A service extensible, completely new service safely-extensible platform that eliminates the need to deny that a processing that might cause a false access is included in an execute form and that can eliminate a false access by an execute form that realizes a digital content providing service. A service safely-extensible platform (SEP), in which the correspondence between a service (S) and an execute form (DE) is established and the service (S) is extended by changing or adding the execute form (DE), has a service-dependent API used for extending the service. The service is extended by the execute form (DE) only through a call of the service-dependent API.

(57) 要約: 実行形式に不正アクセスを引き起こす処理が含まれることを否定する必要無く、デジタルコンテンツの提供サービスを実現する実行形式による不正アクセスの排除できる、サービスの拡張可能な全く新しいサービス安全拡張プラットフォームを提供することを目的とする。サービス (S) と実行形式 (DE) とが対応付けられており、前記実行形式 (DE) の変更や追加によって前記サービス (S) の拡張が達成されるサービス安全拡張プラットフォーム (SEP) は、サービスの拡張を行うサービス依存 API を備え、かつ実行形式 (DE) からのサービスの拡張はサービス依存 API の呼び出しによってのみ行われる。



2文字コード及び他の略語については、定期発行される  
各PCTガゼットの巻頭に掲載されている「コードと略語  
のガイダンスノート」を参照。

## 明 細 書

### サービス安全拡張プラットフォーム

#### 技術分野

デジタルコンテンツの提供サービスを実現する実行形式による不正アクセスの排除を、実行形式に不正アクセスを引き起こす処理が含まれることを否定する必要無く達成するという、新次元の安全性を実現した、サービスの拡張可能な全く新しい「サービス安全拡張プラットフォーム」に関する。詳述すれば、実行形式の配布元の確からしさからの推測を必要とせずに、他のサービスの状態変更、データ破壊、およびプラットフォーム自体のシステムダウンなどに代表される不正アクセスを排除できると共に、実行形式の変更や追加によってサービスの機能／諸元の変更や新規サービスの追加というサービス拡張も達成できるサービス安全拡張プラットフォームに関する。

#### 背景技術

上述の如く、本発明が提供するものは、従来にない全く新しく提供する「サービス安全拡張プラットフォーム」であるので、従来の技術として示すべき適切な例を挙げるのは非常に困難である。それゆえに、先ず、本発明が初めて提唱すると信じるサービス安全拡張プラットフォームが固有に備える3つの主な特徴について以下に述べる。第1の特徴はサービスの拡張性であり、第2の特徴は安全性の確

保であり、そして第 3 の特徴はサービス拡張操作のシームレス化である。

第 1 の特徴である「サービスの拡張性」とは、サービス毎に異なるユーザインタフェースを持ち、ユーザインタフェースの変更や予期せぬサービスの追加が任意の時点で可能であることを言う。第 2 の特徴である「安全性の確保」とは、実行形式に不正アクセスを引き起こす処理が含まれることの否定を必要とせずに安全性が確保されることを言う。そして、第 3 の特徴である「サービス拡張操作のシームレス化」とは、サービス拡張のための操作手順が、サービスの利用時と同じ操作感のもとに行えることを言う。

上述のように、これら 3 つの主な特徴を全て満たす「サービス安全拡張プラットフォーム」を、従来技術において見いだすことはできない。しかしながら、強いて言えば、第 1 の特徴のみを満たすものとして、パーソナルコンピュータを利用してインターネットによるプッシュ型サービスを従来技術の一例として挙げることができる。ポイントキャストネットワーク社のポイントキャスト（R）は、ニュース配信サービスを実現したプッシュ型サービスである。パーソナルコンピュータを利用する複数のプッシュ型サービスの利用は、機械語で記述されたサービス毎に実装したブラウザをパーソナルコンピュータへのインストールすることにより可能となる。

ここで、プッシュ型サービスにおける、本発明の第 1 の特徴である「サービスの拡張性」について説明する。プッシュ型サービスを実現するブラウザをパーソナルコンピュ

ータのハードディスクにインストールすることで新たなサービスを追加することができる。ブラウザは機械語で記述されたコンピュータプログラムであり、サービス毎に固有のそれぞれ異なるユーザインタフェースを提供する。

ブラウザのインストールは、従来、以下の様にして行われる。例えば、ブラウザをインストールする際には、先ず、ファイル転送プロトコル（例：f t p）のクライアントを起動しておき、ブラウザをクライアントのハードディスクなどにダウンロードし、次にダウンロードしたブラウザを起動することで実現される。また、新たなブラウザを利用する前には、ブラウザの設定を、ブラウザに組み込まれた設定メニューで操作して実現する。

次に、本発明の第2の特徴である「安全性の確保」に関しては、従来は単なる憶測に基づくものであって、具体的な方策の裏付けを有するものではない。つまり、従来、ブラウザが安全であるということは、ブラウザが公式ホームページなど正式な配布先より入手したものであるもので、悪意のある第3者がコンピュータウィルスなどを感染させてないはずだという単なる願望に過ぎない。

そして、サービスによる高度な提示機能を意味する本発明の第3の特徴である「サービス拡張操作のシームレス化」に関しては、従来の技術においては、サービス毎にブラウザを変更することで異なるユーザインタフェースを実現する。結果、操作のシームレス化の一部を構成するために必要な、複数サービスに対し異なるユーザインタフェースを持つ要件は満される。

図 1 8 を参照して、上述のようなサービスの拡張は可能であるが安全ではない、従来のサービスプラットフォームについて具体的に説明する。同図に示すように、従来のサービスプラットフォーム P F c は、サービス提供源 1 6 1 0、デリバリシステム 1 2 0、および端末 1 6 3 0 とを含む。サービス提供源 1 6 1 0 は、実行ファイル提供器 1 1 1 1 およびファイル転送サーバ 1 6 1 1 を含む。実行ファイル提供器 1 1 1 1 は実行ファイルを格納したハードディスク装置でよい。また、ファイル転送サーバ 1 6 1 1 は W e b サーバでよい。なお、実行ファイルとは、図 1 9 を参照して後述するように、端末 1 6 3 0 の O S に直接渡して、実行されるプログラムファイルに代表されるバイナリデータである。

デリバリシステム 1 2 0 は、サービス提供源 1 6 1 0 から送られる実行ファイルを空間的または時間的にはなれた端末 1 6 3 0 に向けて伝送する。

端末 1 6 3 0 は、エグゼキュータ 1 1 3 3 および実行ファイル格納器 1 1 3 2 を含む。エグゼキュータ 1 1 3 3 は、実行ファイル格納器 1 1 3 2 に格納された実行ファイルを起動することで端末 1 6 3 0 におけるあらゆる処理を実行する。例えば、実行ファイルとしてファイル転送プログラムを起動した場合には、デリバリシステム 1 2 0 を経由して受信した、新たなサービスを実現する実行ファイルを実行ファイル格納器 1 1 3 2 に格納させる。また、サービスを実現する実行ファイルを実行させれば、サービスをユーザに提供できる。



図 1 9 に、サービスプラットフォーム P F c において、実行ファイル格納器 1 1 3 2 に格納されるデータを例示する。サービス S 1 の実行ファイルである F E ( S 1 ) およびサービス S 2 の実行ファイルである F E ( S 2 ) に並んで、実行ファイルの起動などのプラットフォーム操作を行うユーザインタフェースを提供する s h e l l の実行ファイル F E ( s h e l l ) や、新たなサービスを実現する実行ファイル F E をデリバリシステム 1 2 0 経由で端末 1 6 3 0 に導入するファイル転送プログラムである F E ( ファイル転送 ) が実行ファイル格納器 1 1 3 2 内に格納されている。これらの実行ファイルは何れも機械語で実装される。

図 2 0 に、サービスプラットフォーム P F c のソフトウェア階層を示す。同図から読みとれるように、サービスプラットフォーム P F c において、ファイル転送実行ファイルを起動することで、あらゆるサービスを実現する実行ファイルを新たに導入できる。このように、O S ( O p e r a t i n g S y s t e m ) が構成する O S 層の直上に、実行ファイルによってアプリケーション層が構成されている。結果、サービスを実現する実行ファイルが直接 O S のリソースを参照したり操作したりできる。それゆえに、O S は、悪意を持った実行ファイルに対して無防備であるので、サービスプラットフォーム P F c はとうてい安全とは言えない。

また、本発明の第 1 の特徴である「サービスの拡張性」に関しては、新たなサービスの単純な導入は可能であるが

、本発明が提供するコンテンツの視聴とシームレス化した新たなサービスの導入については、従来は不可能である。つまり、コンテンツの視聴と新たなサービスの導入とにおける操作性は、その見かけも手順も全く異なる。例えば、従来は、ブラウザをインストールする際には、ファイル転送プロトコルのクライアントを起動し、ブラウザをクライアントのハードディスクなどにダウンロードし、次にダウンロードしたブラウザを起動しなければならない。また、新たなブラウザを利用する前には、ブラウザの設定を、ブラウザによるコンテンツの視聴画面とは全く別の使い勝手の設定メニューを操作しなければならない。

また、従来は、全てのサービスに対して、共通のユーザーインタフェースを利用して機能拡張が行われる。そのために、サービス拡張するために、プログラムをインストールする際に必要な各種パラメータ（インストール先のディレクトリやユーザー名等）の入力が、全サービスで必要なパラメータの和集合となってしまう。そのために、特定のサービスの拡張プログラムをインストールするために、本来不要なパラメータの入力まで、ユーザーに求めてしまうことがある。

一方、本発明にかかるサービス安全拡張プラットフォームが提供するサービスの柔軟な拡張性は、ブラウザを終了したり、コンテンツの視聴と全く異なる使い勝手の手順を踏むことなく、あくまでもコンテンツの視聴と同じ操作の使い勝手の中でサービスの拡張を行えることである。すなわち、サービスの拡張のためだけの特別な使い勝手の操作



をユーザに強要することなく、ユーザが最も慣れているであろうコンテンツの視聴と同じ使い勝手で、あたかも、通常の操作の延長として、サービスが拡張されていく状況を達成するものである。しかしながら、このように拡張性は、従来のサービスプラットフォームの提供できることではないことは上述の通りであり、そのために、全く新しいサービス拡張の実現方法が求められている。

次に、本発明の第2の特徴である「安全性の確保」に関しても、正式な配布先から入手したブラウザでもバグによって、他のプッシュ型サービスの動作に悪影響を与えたり、システム全体をダウンさせたりすることも少なくないのが現実である。そのため、従来においては、ブラウザのバイナリに正当性がある、すなわち、ブラウザのバイナリに不正アクセスの処理が含まれないであろうことを少なくとも保証する必要がある。しかし、その保証はあくまでも信頼関係に基づくものであって、ブラウザのバイナリに不正アクセス処理が含まれていないことを確実にするものではない。

一方、本発明にかかるサービス安全拡張プラットフォームが提供する安全性は、如何なる実行形式（従来例でのブラウザに対応）の実行によっても、他のサービスへの悪影響や、システム全体のダウンを起こさない、全く新しい次元の安全性である。当然、従来の技術においては、このように方式は存在せず、それゆえに全く新しい実現方法が求められているものである。

さらに、本発明の第3の特徴である高度な提示機能を実

現する「サービス拡張操作のシームレス化」に関しても、従来の技術ではサービス管理とコンテンツの視聴とのシームレス化は不可能である。その原因は、サービス管理はブラウザの設定メニューで行われていることにある。つまり、サービスの管理に関わることは、コンテンツの視聴画面では行えないのである。

一方、本発明にかかるサービス安全拡張プラットフォームが提供するサービスによる高度な提示機能は、コンテンツにサービスの設定を含めることで、コンテンツ視聴と同一の使い勝手の中で、コンテンツの持つ高度な表現を持ち、かつ、サーバから自由に提示内容を設定することを可能とする。

例えば、アンケートに記述しサーバに向けて回答を返信するといった、高度な記述能力を使用したコンテンツにサービスの設定機能を組み合わせることで、アンケートにユーザが興味にある内容を回答することで、自動的にコンテンツの取捨選択を行うためのユーザの嗜好情報が設定されるといった高度な提示機能を実現できる。また、ユーザにとっては、どこまでがコンテンツなのか、サービスの設定なのかを全く意識させることなく、高度な設定を簡単な操作で行える。

当然、従来の技術においては、本発明にかかるサービス安全拡張プラットフォームが提供するサービスによる高度な提示機能を実現する方式が存在せず、それゆえに全く新しい実現方法が求められているものである。

よって、本発明は、上述の3つの特徴を実現するサービ

ス安全拡張プラットフォームを提供することを目的とする。

### 発明の開示

本発明は、上記のような目的を達成するために、以下に述べるような特徴を有している。

第1の局面は、サービスと実行形式とが対応付けられており、実行形式の変更や追加によってサービスの拡張が達成されるサービス安全拡張プラットフォームであって、サービスの拡張を行うサービス依存APIを具備し、かつ実行形式からのサービスの拡張はサービス依存APIの呼び出しによってのみ行われることを特徴とする。

上述のように、第1の局面においては、実行形式の正当性に依らない安全性が確保できる。

第2の局面は、第1の局面において、サービスの拡張が新規サービスの新設であることを特徴とする。

第3の局面は、第1の局面において、サービスの拡張がサービス利用開始であることを特徴とする。

第4の局面は、第1の局面、第2の局面、および第3の局面の何れかにおいて、注目するサービスの拡張は、注目するサービスに対応付けられた実行形式からのサービス依存APIの呼び出しによってのみ行われることを特徴とする。

第5の局面は、第1の局面、第2の局面、および第3の局面の何れかにおいて、複数のサービス間に親子関係が定義され、実行形式が要求するサービス依存API呼び出し

がサービスに対応付けられるサービス依存リソースを処理対象として指定した際に、実行形式に対応付けられるサービスがサービスの先祖である場合にのみ、サービス依存リソースに対して処理可能であることを特徴とする。

第6の局面は、第1の局面、第2の局面、および第3の局面の何れかにおいて、メタサービスに対応付けられた実行形式がサービス依存APIによって、サービスの少なくとも1つが拡張可能なことを特徴とする。

第7の局面は、第6の局面において、メタサービスに対応付けられた実行形式がサービス依存APIによって、サービスの全てが拡張可能であり、メタサービスに対応付けられていない実行形式(DE)はサービス依存APIによってサービスの拡張が不可能であることを特徴とする。

第8の局面は、第1の局面、第2の局面、および第3の局面の何れかにおいて、実行形式がコンテンツとして満たすべき条件を満たした制御コンテンツであり、制御コンテンツ(DC)がコンテンツの少なくとも1つと共にコンテンツとして伝送され、コンテンツ(DC)の少なくとも1つから制御コンテンツを指定する情報が伝送され、制御コンテンツによってのみサービス依存APIの処理が可能であることを特徴とする。

第9の局面は、第8の局面において、サービス依存APIによって、特定のサービスのコンテンツ(DC)の自動的な格納が制御されることを特徴とする。

第10の局面は、第1の局面、第2の局面、第3の局面、第4の局面、第5の局面、第6の局面、第7の局面、第

8の局面、および第9の局面の何れかにおいて、実行形式を少なくとも1つのサービス提供部に送出し、実行形式を実行する少なくとも1つの端末で受信することを特徴とする。

なお、本局面の一実施の形態においては、ブラウザを終了したり、コンテンツの視聴と全く異なる使い勝手の手順を踏むことなく、あくまでコンテンツの視聴と同じ操作の使い勝手の中でサービスの拡張を行える。すなわち、サービスの拡張のためだけの特別の使い勝手の操作をユーザに強要することなく、ユーザが最も慣れているであろうコンテンツの視聴と同じ使い勝手のうえで、いわば、いつの間にかサービスが拡張されている状況を達成できる。

また、如何なる実行形式（従来例でのブラウザに対応）の実行によっても、他のサービスへの悪影響や、システム全体のダウンを起こさない、全く新しい次元の安全性を達成できる。

さらに、異なる実施の形態においては、コンテンツにサービスの設定を含めることで、コンテンツ視聴と同一の使い勝手の中で、コンテンツの持つ高度な表現を持ち、かつ、サーバから自由に提示内容を設定できる機能である。そして、ユーザにとっては、どこまでがコンテンツなのか、サービスの設定なのかを全く意識させることなく、高度な設定を簡単な操作で行える。

#### 図面の簡単な説明

図1は、本発明の第1の実施の形態にかかる、サービス

安全拡張プラットフォームの構成を模式的に示すブロック図である。

図 2 は、図 1 に示したサービス安全拡張プラットフォームの識別化実行形式格納器に格納される識別化実行形式を示す説明図である。

図 3 は、図 1 に示したサービス安全拡張プラットフォームのサービス依存リソース管理者で管理されるサービス依存リソース管理テーブルの説明図である。

図 4 は、図 1 に示したサービス安全拡張プラットフォームのサービス依存リソース管理者による新サービスの登録管理動作を示すフローチャートである。

図 5 は、図 1 に示したサービス安全拡張プラットフォームの端末によるサービス実行動作を示すフローチャートである。

図 6 は、図 1 に示したサービス安全拡張プラットフォームの端末のソフトウェア階層を示す説明図である。

図 7 は、図 1 に示したサービス安全拡張プラットフォームによってユーザに提示される画面例を示す説明図である。

図 8 は、本発明の第 2 の実施の形態にかかるサービス安全拡張プラットフォームの構成を模式的に示すブロック図である。

図 9 は、図 8 に示したサービス安全拡張プラットフォームの識別化実行形式格納器に格納される識別化実行形式およびメタサービス実行形式を示す説明図である。

図 10 は、図 8 に示したサービス安全拡張プラットフォーム



ームの端末によるサービス実行動作を示すフローチャートである。

図 1 1 は、本発明の第 3 の実施の形態にかかるサービス安全拡張プラットフォームの構成を模式的に示すブロック図である。

図 1 2 は、図 1 1 に示したサービス安全拡張プラットフォームの識別化実行形式格納器に格納される識別化実行形式および親子識別化実行形式を示す説明図である。

図 1 3 は、図 1 1 に示したサービス安全拡張プラットフォームの端末によるサービス実行動作を示すフローチャートである。

図 1 4 は、本発明の第 4 の実施の形態にかかるサービス安全拡張プラットフォームの構成を模式的に示すブロック図である。

図 1 5 は、図 1 4 に示したサービス安全拡張プラットフォームのコンテンツ格納器に格納される制御コンテンツおよびコンテンツを示す説明図である。

図 1 6 は、図 1 4 に示したサービス安全拡張プラットフォームのサービス依存リソース管理者で管理されるサービス依存リソース管理テーブルの説明図である。

図 1 7 は、図 1 4 に示したサービス安全拡張プラットフォームの端末による制御コンテンツ実行動作を表すフローチャートである。

図 1 8 は、従来の技術における、従来のサービスプラットフォームの構成を模式的に示すブロック図である。

図 1 9 は、図 1 8 に示したサービスプラットフォームの

実行ファイル格納器に格納される実行ファイルを示す説明図である。

図 20 は、図 18 に示したサービスプラットフォームの端末におけるソフトウェア階層を示す説明図である。

### 発明を実施するための最良の形態

先ず、本発明にかかるサービス安全拡張プラットフォームの基本的概念について説明する。本発明にかかるサービス安全拡張プラットフォームにおいて、サービスは、従来のような OS によって直接実行される、例えばバイナリの実行ファイルではなく、OS に対する実行を指示するエクセキュータにより解釈されるいわばスクリプトとして構成される実行形式として構成される。そして、実行形式は、API 呼び出しを含み、所定のサービスに対応付けられる。

API は、呼び出し時に、対応付けられたサービスに依存した挙動をするサービス依存である。サービス依存 API は、含まれる実行形式に対応付けられたサービスに固有のリソースに対して処理する。また、実行形式からリソースへのあらゆる処理は API 経由に限定され、かつ、サービス依存 API により処理されるリソースに対して処理を行うサービス依存 API 以外の API は存在しない。また、実行形式内に記述する API のセットは、全てのサービスに対し共通に予め用意されたものであり、同一である。

このように構成することによって、従来における実行ファイル（あるいは本発明における実行形式）の正当性に依

らない新次元の安全性を確保し、サービスの拡張が可能な全く新しい「サービス安全拡張プラットフォーム」を創出する。同プラットフォームにおいては、サービスを実現する実行形式の一例であるブラウザを動作させることで、サービスのインスタンスであるコンテンツが視聴できる。コンテンツの通常の視聴は、コンテンツに含まれるAPIを逐次的に呼び出して実現される。そのために、実行形式（ブラウザ）は、APIの一種であるサービス依存APIを含んで構成される。つまり、コンテンツの視聴の最中に、サービス依存APIが呼び出されることで、サービスが拡張される。結果、本発明においては、サービスの拡張は、ユーザにはコンテンツの視聴操作の一つとして見える。

サービスを拡張するユーザインタフェースは、サービス依存APIを含む実行形式によって決定される。実行形式をサービスに対応付けることで、個々のサービスのそれぞれに適したユーザインタフェースを利用してサービスが拡張できる。

サービス拡張のユーザインタフェースを、サービス毎に最適に構成できる。例えば、プッシュ型サービス／蓄積型放送サービスの場合、同サービスの会員の中で、特定の会員のみ加入可能なプレミアムサービスの契約を、通常のサービスのコンテンツを利用してそのような特定の会員のみに対して勧誘できる。つまり、目的などに応じてユーザインタフェースを必要十分にカスタマイズできるので、より適切且つ強力にサービスの勧誘が出来る。

携帯電話等で利用されるJavaアプリ（iアプリ等）

に本発明を適応すれば、ユーザが購読しているサービスを実現するアプリケーションソフトウェアに、サービスの機能を拡張するプログラムを記述することで、同サービスの機能拡張を実現できる。例えば、スキンなどのiアプリの機能をプラグイン的に拡張する操作をiアプリ自身で実現できる。

更には、コンテンツ内に含まれるスクリプトに条件判断として、問いかけに回答して受信機の型番やメーカーの種別を表す文字列を返すAPIを呼び出すようにコンテンツを記述することで、受信機の型番やメーカー毎にユーザインタフェースを変化させることができる。

また、デジタル放送におけるペイパービュー番組に適応すれば、配信コンテンツに適切なアルゴリズム記述することで、おためし無料視聴が可能な期間やチャンネルをコンテンツにプログラミングしたアルゴリズムによって管理できる。デジタル放送でのペイパービュー番組の場合、ペイパービューのチャンネルのデータ放送を通してペイパービューの契約ができる。また、データ放送のゲームで高得点をあげたユーザに対してのみ安価な料金でサービス契約するというような、契約促進に繋がる、ユーザ毎に異なるサービスの提供が可能である。

次に、コンテンツに、サービス依存APIが実行されるようにプログラミングすることによって、コンテンツの記述能力を損なうことなく、ユーザがコンテンツに対して所定の操作を行うことによって、コンテンツに関してシームレスにサービス拡張ができる。

例えば、Webブラウザのplug-inをインストールする場合には、同plug-inのインストールプログラムを入手する為に、同plug-in開発ベンダーのダウンロードページにアクセスせずに、コンテンツの受信時にPlug-inをインストール出来る。また、WebブラウザのPlug-inの商品名について、ユーザが熟知しておく必要はなく、「商品を360度の角度からみた動画」というような、コンテンツ本意の説明を選択するだけで実行できる。

また、電子ブックプレーヤやPDAに適応すれば、特定ホームページに入った時に同ホームページで必要とされる掲示板表示専用ブラウザなどが、ユーザが気づかないうちに自動的に使えるように準備される機能を提供できる。いわば、コンテンツの視聴とサービス拡張に用いられるユーザインタフェースをシームレスに構成出来る。コンテンツ操作に於けるのと同じ操作性のユーザインタフェースを利用してサービス拡張ができるので、ユーザにかかる負担が小さい。

次に、逐次的に実行されるAPIとして、サービス依存APIを追加することによって、実行形式を実行させれば、サービス依存APIも他のAPI同様逐次的に実行される。結果、自動でサービスの拡張を実行するように実行形式をプログラミングすることも可能である。また、完全に自動化することで、旅行中の専用サービスや無料試用など、その時限りのサービス拡張もユーザが無意識のうちに実現できる。

例えば、電子ブックプレーヤーやPDAの場合、ページめくりの際の画面効果をタイトルに付随し、そのタイトルでしか使えないプレミアムなプラグインとして提供できる。また、成田空港内のバーチャル施設案内など、その場でしか意味のないサービスを自動で利用できる。

次に、本発明においては、ブラウザを終了したり、コンテンツの視聴と全く異なる使い勝手の手順を踏むことなく、あくまでもコンテンツの視聴と同じ操作の使い勝手の中でサービスの拡張が行えることを保証する。つまり、サービスの拡張のためだけに用意される特別の使い勝手の操作をユーザに強要することなく、ユーザが最も慣れているであろうコンテンツの視聴と同じ使い勝手で、ユーザがそれと特に意識することなく、気が付けばサービスが拡張されている環境を提供する。

例えば、プッシュ型サービス／蓄積型放送サービスの場合、ビューワのバージョンアップならそのビューワ自体のアップグレード機能、別のサービスの導入ならWebブラウザを起動してダウンロードといった複数のユーザインタフェースの作法に慣れている必要がない。

つまり、コンテンツの内容や、ユーザの好みに応じて、複数のブラウザを使い分ける必要のある場合にも、ユーザは様々なユーザインタフェースの異なる操作性に慣れる必要がない。それ故に、不慣れなユーザでも安心し、且つ容易に、機能拡張できる。そして、視聴装置の出荷後に、ユーザインタフェースが固定化されることなく、ユーザの使い勝手の応じて、その操作性を改善出来る。



更には、全てに共通するが、如何なる実行形式（従来例でのブラウザに対応）の実行によっても、他のサービスへの悪影響や、システム全体のダウンを起こさない、全く新しい次元の安全性が確保できる。

（第 1 の実施の形態）

図 1、図 2、図 3、図 4、図 5、図 6 および図 7 を参照して、本発明の第 1 の実施の形態にかかるサービス安全拡張プラットフォームについて説明する。図 1 に示すように、本実施の形態にかかるサービス安全拡張プラットフォーム S E P 1 は、サービス提供源 1 1 0、デリバリシステム 1 2 0、および端末 1 3 0 を含む。

サービス提供源 1 1 0 は、端末 1 3 0 が実行すべき新たなサービスを実現するための実行形式を送出する。デリバリシステム 1 2 0 は、サービス提供源 1 1 0 が送出的情報を出する情報を端末 1 3 0 に向けて時間的かつ／あるいは空間的に移動させる。デリバリシステム 1 2 0 は、インターネット通信網、放送や通信の無線ネットワーク、あるいは、D V D - R O M ( D i g i t a l V e r s a t i l e D i s k - R e a d O n l y M e m o r y ) などのパッケージメディアと物理的な物流システムの組み合わせで構成できる。

端末 1 3 0 は、~~デリバリシステム 1 2 0 経由で受け取った~~情報をを用いて、サービス提供源 1 1 0 から提供されたサービスを実行して、実行結果をユーザに提供する。なお、図 1 においては、簡便化のために、サービス提供源 1 1 0、デリバリシステム 1 2 0、および端末 1 3 0 のそれぞれ

の台数比が 1 : 1 : 1 であるように表されている。しかしながら、台数比は放送の形態に類似した 1 : 1 : c (c は任意の自然数) や、インターネットなどの a : 1 : c (a は任意の自然数) や、あるいは複数のデリバリシステムを持つ場合の a : b : c (b は任意の自然数) であってもよい。このような、一般化は以下に述べる全ての実施の形態において当てはまる。

サービス提供源 1 1 0 は、実行形式提供器 1 1 1、サービス識別設定器 1 1 2、および送出器 1 1 3 を含む。実行形式提供器 1 1 1 は、サービスを実現する実行形式を格納し、必要に応じて出力する。この実行形式は、インターネットで用いられる HTML 言語 (Hyper Text Markup Language) や、日本のデジタル放送のデータ放送で用いられる BML 言語 (Broadcast Markup Language) などの、SGML (Standard Generalized Markup Language) / XML (eXtensible Markup Language) 系のマークアップ言語や、仮想マシン上で動作する Java (R) 言語、あるいは機械語などでよい。ただし、実行形式は、後述する端末 1 3 0 に備えられたエグゼキュータを介して、端末 1 3 0 の OS に対して渡されて実行されるように構成される。

サービス識別設定器 1 1 2 は、実行形式提供器 1 1 1 の出力する実行形式 DE に対して、実行形式 DE の属性等の副次的な事項を表す付随情報 IS を生成すると共に実行形

式 D E に付与して識別化実行形式 D E i を生成する。付随情報 I S は、対応する実行形式 D E が実現するサービスとを対応付ける情報であるサービス識別情報 E s と、その実行形式 D E の使用条件、使用状態、および対応するサービス内容などの情報を表す副次情報  $\alpha$  を含む。サービス識別情報 E s は、例えば、重複しない数字である I D や名称などでよい。副次情報  $\alpha$  はテキストでもコードでもよい。また、識別化実行形式 D E i は、実行形式 D E と付随情報 I S を一体的に生成してもよいし、互いに独立して生成してもよい。本実施の形態においては、視認性を考慮して付随情報 I S は独立して生成される場合を例に説明する。

送出器 1 1 3 は、サービス識別設定器 1 1 2 から入力される識別化実行形式 D E i (実行形式 D E および付随情報 I S) をデリバリシステム 1 2 0 に送出する。この送出を実現する伝送モデルは、いわゆるプル型およびプッシュ型の何れでもよい。

プル型とは、インターネットのホームページ閲覧に用いられる伝送プロトコルである H T T P (H y p e r T e x t T r a n s p o r t P r o t o c o l) などで見られるように、受信側である端末 1 3 0 からの要求 (d e m a n d) に基づき送出する伝送モデルである。またプッシュ型とは、デジタル放送の伝送に用いられる伝送プロトコルである D S M - C C (D i g i t a l S t o r a g e M e d i a C o m m a n d & C o n t r o l) データカルーセルなどで見られるように、受信側の要求に関わらず所定のタイミングで送出側から送出する伝送モデル

である。

端末 1 3 0 は、ダウンローダ 1 3 1、識別化実行形式格納器 1 3 2、エグゼキュータ 1 3 3、リソースセクタ 1 3 4、一般リソース管理者 1 3 5、およびサービス依存リソース管理者 1 3 6 を含む。ダウンローダ 1 3 1 は、デリバリシステム 1 2 0 から伝送されてくる、識別化実行形式 D E i（実行形式 D E および付随情報 I S）を受信し、受信した識別化実行形式 D E i を識別化実行形式格納器 1 3 2 に書き込むと共に、識別化実行形式 D E i から付随情報 I S を抽出してサービス依存リソース管理者 1 3 6 に出力する。付随情報 I S は、上述のようにサービス識別設定器 1 1 2 で設定されたサービス識別情報 E s を含む。ダウンローダ 1 3 1 は、送出器 1 1 3 とデリバリシステム 1 2 0 で実現される伝送モデルに整合しており、プッシュ型でもプル型でも実施可能である。

識別化実行形式格納器 1 3 2 は、ダウンローダ 1 3 1 により書き込まれた識別化実行形式 D E i（実行形式 D E および付随情報 I S）を格納する。また、識別化実行形式格納器 1 3 2 は、要求に応じて格納した識別化実行形式 D E i をエグゼキュータ 1 3 3 に出力する。識別化実行形式格納器 1 3 2 はハードディスクドライブ（H D D）や D V D - R A M などの記録媒体や、フラッシュメモリや R A M などの半導体メモリを用いて構成できる。

図 2 に、本実施の形態において、識別化実行形式 D E i が識別化実行形式格納器 1 3 2 に格納される様子を示す。同例において、サービス S 1、サービス S 2 乃至サービス

$S_n$  ( $n$  は任意の自然数) に関して、識別化実行形式  $DE_i$  ( $S_1$ ) はサービス  $S_1$  に対応し、識別化実行形式  $DE_i$  ( $S_2$ ) はサービス  $S_2$  に対応し、識別化実行形式  $DE_i$  ( $S_n$ ) はサービス  $S_n$  に対応する。

そして、識別化実行形式  $DE_i$  ( $S_1$ ) は、それぞれサービス  $S_1$  に対応するサービス識別情報  $E_s$  ( $S_1$ ) と副次情報  $\alpha_1$  から成る付随情報  $IS$  ( $S_1$ ) と実行形式  $DE$  を含む。具体的には、サービス識別情報  $E_s$  ( $S_1$ ) が実行形式  $DE$  とサービス  $S_1$  とを対応付けている。同様に、識別化実行形式  $DE_i$  ( $S_2$ ) は、それぞれサービス  $S_2$  に対応するサービス識別情報  $E_s$  ( $S_2$ ) と副次情報  $\alpha_2$  から成る付随情報  $IS$  ( $S_2$ ) と実行形式  $DE$  を含む。さらに、識別化実行形式  $DE_i$  ( $S_n$ ) は、サービス  $S_n$  に対応するサービス識別情報  $E_s$  ( $S_n$ ) と副次情報  $\alpha_n$  から成る付随情報  $IS$  ( $S_n$ ) と実行形式  $DE$  を含む。実行形式  $DE$  はサービス識別情報  $E_s$  ( $S_n$ ) によってサービス  $S_2$  に対応付けられている。

サービスを個々に区別する必要のない場合、識別化実行形式  $DE_i$  はサービス識別情報  $E_s$  および副次情報  $\alpha$  から成る付随情報  $IS$  と実行形式  $DE$  で構成されると表現する。そして、サービスを個々に区別する必要がある場合は、識別化実行形式  $DE_i$  ( $S_o$ ) は、サービス識別情報  $E_s$  ( $S_o$ ) および副次情報  $\alpha_o$  から成る付随情報  $IS$  ( $S_o$ ) と実行形式  $DE$  で構成されると表現する。なお、 $o$  は  $n$  以下の任意の自然数である。

図 1 に戻って、エグゼキュータ 133 は、識別化実行形

式格納器 1 3 2 から入力される識別化実行形式 D E i に含まれる実行形式 D E を解釈してサービスを実行する。ただし、識別化実行形式格納器 1 3 2 に格納された識別化実行形式 D E i に含まれる実行形式 D E は、エグゼキュータ 1 3 3 以外では端末 1 3 0 内で解釈実行されることはない特徴を有していることは上述の通りである。エグゼキュータ 1 3 3 は、また、ユーザからのキーボードやポインティングデバイスや音声入力デバイスといった入力デバイスからの入力 I u や、G U I の画面表示や音声出力などの出力デバイスへの出力、といった対話処理を実行形式 D E に基づいた手順で実現する。

エグゼキュータ 1 3 3 は、実行形式 D E が J a v a ( R ) 言語やマークアップ言語であれば仮想マシンや、一般にはブラウザと呼ばれる実行環境でよい。また、実行形式 D E が機械語であれば、それを実行させるための O S に付属するライブラリなどのミドルウェア群などである。

エグゼキュータ 1 3 3 は、実行形式 D E を解釈実行していく際に、実行形式 D E に A P I ( A p p l i c a t i o n P r o g r a m I n t e r f a c e ) 呼び出し C a p i が含まれている場合には、リソースセクタ 1 3 4 を経由して一般リソース管理者 1 3 5 もしくはサービス依存リソース管理者 1 3 6 に対して A P I 呼出 C a p i を発行して、処理の実行を要求する。また、エグゼキュータ 1 3 3 は、現在実行している実行形式 D E に対応付けられたサービス識別情報 E s をサービス依存リソース管理者 1 3 6 に通知する。



リソースセクタ 134 は、エグゼキュータ 133 から発行された A P I 呼出 C a p i に基づいて、一般リソース管理者 135 およびサービス依存リソース管理者 136 の何れに対して、A P I 処理が要求されているのかを判断する。そして、要求されていると判断されている方に、A P I 呼出 C a p i を伝達する。なお、A P I 呼出 C a p i には、一般リソース管理者 135 が処理すべき一般の A P I（以後、「一般 A P I」と呼ぶ）と、サービス依存リソース管理者 136 が処理すべき A P I（以降、「サービス依存 A P I」）とに分類される。

リソースとは、端末 130 がエグゼキュータ 133 を経由して、参照、変更、あるいは制御可能なあらゆる計算機資源を言う。すなわち、R A M への 1 次記録器や H D D などの 2 次記録器に格納されるデータ構造、制御可能な入出力デバイスに対するアクセス権と具体的な入出力制御、および通信制御などが含まれる。これらリソースのうち、他のサービスへの影響を与えることなく処理可能なリソースを一般リソースとし、サービス毎に存在し、かつサービスの拡張の際に処理が必要なリソースをサービス依存リソースと定義する。そして、一般リソースに対する A P I を一般 A P I とし、サービス依存リソースに対する A P I をサービス依存 A P I と定義する。

さらに、2つの A P I と同一のリソースとの関係に例をおいて説明する。そして、一方の A P I はリソースを参照して他のサービスに影響しないが、他方の A P I はリソースを変更して他のサービスに影響を及ぼすと想定する。こ

の場合、前者が一般 A P I に対応し、後者がサービス依存 A P I に対応する。しかしながら、説明の簡便化のために、参照のためのリソースと変更のためのリソースとがそれぞれ独立に存在すると見なして、前者と後者の内容が整合されていると捉えて説明する。

サービス依存 A P I の具体例としては、サービスの利用状態を変更する関数（以後、「サービス利用状態操作関数」と称する）などが考えられる。サービス利用状態操作関数によって、利用者が端末 1 3 0 で個別のサービスを利用するのかわしないのかを指定し、サービスを提供する際に端末 1 3 0 が行うべき処理の起動と停止などを制御する。具体的には実行形式 D E あるいは識別化実行形式 D E i の内部データの初期化や必要な情報の受信などの前処理や、デリバリシステムを経由したサービス提供部に対するサービス利用契約の締結、課金、およびユーザ登録などである。

一方、一般 A P I の具体例としては、端末受信部 1 3 0 の画面表示やキーボード入力といった入出力デバイスへの操作や、R A M への一時記憶や H D D への 2 次記憶に対するデータの読み書きなどがある。ただし、一般 A P I のリソースである画面表示を例にした場合に、複数のサービスに対応する画面表示が同時に出現する際には、サービス間で競合が発生することとも考えられる。しかしながら、本実施の形態においては、実際には同時には 1 つのサービスのみが画面を占有するという制約を一般リソース管理者 1 3 5 や A P I の呼び出し方法などで実現すれば、あらゆる実行形式 D E に対しても実際には競合が発生しない。

一般リソース管理者 1 3 5 は、一般リソースを格納および管理する。一般リソース管理者 1 3 5 は、さらに、リソースセクタ 1 3 4 から入力される A P I 呼出 C a p i に基づいて、一般リソースへの参照や操作などを行う。例えば、画面描画の A P I が呼び出されると画面描画を行う一般リソースであるグラフィック表示デバイスに対して命令を発する。

サービス依存リソース管理者 1 3 6 は、サービス依存リソースを格納および管理する。リソースセクタ 1 3 4 から入力される A P I 呼出 C a p i に基づいて、サービス依存リソース R S への参照や操作などを行う。また、ダウンロード 1 3 1 から入力される付随情報 I S に基づいて、サービス依存リソース R S の管理処理を行う。

図 3 を参照して、サービス依存リソース管理者 1 3 6 に格納されるサービス依存リソース R S を管理するために生成されるサービス依存リソース管理テーブル T r s について説明する。同図に例示するように、サービス依存リソース管理テーブル T r s は、少なくとも、n 種類のサービス S 1 ~ S n を表す複数の行 L 1 ~ L n と、サービス S 毎の利用状態を表す 2 列 C 1 および C 2 からマトリックス状に構成されるデータベースである。具体的には、同図において行 L 1 がサービス S 1 に対応し、行 L 2 がサービス S 2 に対応し、行 L n がサービス S n (この場合、n は 3 以上の自然数) に対応している。そして、列 C 1 はサービス識別情報 E s に対応し、列 C 2 は利用状態に対応する。

なお、より具体的に言えば、列 C 1 および列 C 2 の値は

、それぞれ図 2 に模式的に表したように、サービス識別情報  $E_s(S_n)$  の  $S_n$  および副次情報  $\alpha_n$  に基づいて決定されて書き込まれる。図 3 に示す例においては、行  $L_1$  に示されるサービス  $S_1$  は「未利用」であり、行  $L_2$  に示されるサービス  $S_2$  は「利用」状態であり、行  $L_n$  に示されるサービス  $S_n$  は「利用」状態であることが分かる。このように行  $L_1 \sim L_n$  のそれぞれは、異なるサービス  $S_1 \sim S_n$  のサービスを識別する情報を蓄えるために設けられている。この意味において、行  $L_1 \sim L_n$  (行  $L_0$ ) をサービス識別行と呼ぶ。

次に、新たなサービス識別行を追加する際の動作について説明する。ダウンロード 131 は、サービス(識別化実行形式  $DE_i$ )を受け取ると、識別化実行形式格納器 132 に格納されている実行形式  $DE$  が対応するサービス依存リソース  $RS$  を規定するサービス識別行がサービス依存リソース管理テーブル  $Trs$  に追加される。これらの処理によって、自動的にサービスが端末 130 に導入されることによって、新サービスがサービス依存リソース管理者 136 で登録・管理される。

以下に、図 4 を参照して、サービス依存リソース管理者 136 による新サービスの登録管理ルーチンの動作について説明する。新たなサービスに対応する識別化実行形式  $DE_i$  (実行形式  $DE$  および付随情報  $IS$ ) は、サービス提供源 110 から端末 130 に送出されて、先ずダウンロード 131 に入力される。ダウンロード 131 は、識別化実行形式  $DE_i$  から付随情報  $IS$  を抽出して、サービス依存

リソース管理者 1 3 6 に出力する。そして、サービス依存リソース管理者 1 3 6 においては、付随情報 I S が入力されて時点で、本ルーチンの動作が開始される。

そのため、ステップ S 5 0 2 において、付随情報 I S がサービス依存リソース管理者 1 3 6 に入力されているか否かが判断される。入力されていない場合、N o と判断されて、本ステップにおける処理が繰り返される。一方 Y e s の場合、処理は次のステップ S 5 0 4 に進む。つまり、サービス依存リソース管理者 1 3 6 は、ダウンローダ 1 3 1 から付随情報 I S を受け取るまではサインサービスの登録管理処理は、実質的に開始されない。

ステップ S 5 0 4 において、ステップ S 5 0 2 で受け取った付随情報 I S のサービス識別情報 E s が示すサービス S が、サービス依存リソース管理者 1 3 6 に格納されているサービス依存リソース管理テーブル T r s に既に登録されているか否かが判断される。含まれない、つまり、未登録サービスである場合は、N o と判断されて、処理は次のステップ S 5 0 6 に進む。

ステップ S 5 0 6 において、サービス依存リソース管理者 1 3 6 において、サービス依存リソース管理テーブル T r s に新サービスに対応する新サービス識別行 ( L n + 1 ) が追記される。以降、図 3 に示すサービス依存リソース管理テーブル T r s を例として説明する。サービス依存リソース管理テーブル T r s には、既にサービス S 1 ~ S n ままで登録されているので、新たなサービス S ( n + 1 ) を登録するために、サービス識別行 L ( n + 1 ) が追加さ



れる。そして、処理は次のステップ S 5 0 8 に生成される。

ステップ S 5 0 8 において、新たに受領した付随情報 I S に含まれるサービス識別情報 E s に基づいて、行 L ( n + 1 ) 列 C 1 にサービス S ( n + 1 ) を識別する値である S ( n + 1 ) が記入される。そして、処理は次のステップ S 5 1 0 に進む。

ステップ S 5 1 0 において、サービス依存リソース管理テーブル T r s に追加された行 L ( n + 1 ) 列 C 2 の利用状態の欄に「未利用」が記入される。これは、新規サービス（識別化実行形式 D E i ）が受信された際の初期状態としては、「未利用」を設定するようにしているからである。しかしながら、初期状態が「利用」であったり、一定期間の間デモとして試用するなどの設定にしてもよいし、また、これらの初期状態の何れを採るか示す情報を、付随情報 I S （副次情報  $\alpha$  ）としてサービス識別設定器 1 1 2 によってサービス毎に与えてもよい。本ステップの処理の終了後、本ルーチンを終了する。

一方、上述のステップ S 5 0 4 において、Y e s 、つまり含まれると判定された場合、上述のステップ S 5 0 6 、ステップ S 5 0 8 およびステップ S 5 1 0 をスキップして、本ルーチンにおける処理を終了する。すなわち、ダウンロード 1 3 1 が受け取ったサービスが既に導入されているサービスである場合には、サービス登録は不要であるので、本ルーチンの処理が直ちに終了される。

次に、図 5 を参照して、端末 1 3 0 によるサービス実行



ルーチンの動作について説明する。具体的には、端末 1 3 0 において、エグゼキュータ 1 3 3 が呼び出す A P I に対する実行形式 D E が実行されることによって、サービス実行が実現される。つまり、エグゼキュータ 1 3 3 が識別化実行形式格納器 1 3 2 から入力される識別化実行形式 D E i に含まれる実行形式 D E を実行させるために、A P I 呼出 C a p i を発行した時点で、本ルーチンにおける実質的処理が開始される。

よって、ステップ S 5 1 2 において、リソースセクタ 1 3 4 によって、エグゼキュータ 1 3 3 から発行された A P I 呼出 C a p i に基づいて、呼び出された A P I がサービス依存 A P I であるか否かが判断される。サービス依存 A P I であれば、Y e s と判断されて、処理は次のステップ S 5 1 4 に進む。

ステップ S 5 1 4 において、サービス依存リソース管理器 1 3 6 によって、現在実行している実行形式 D E に対応付けられたサービスのサービス識別情報 E s を、エグゼキュータ 1 3 3 から得る。そして、処理は、次のステップ S 5 1 6 に進む。

ステップ S 5 1 6 において、サービス依存リソース管理器 1 3 6 によって、A P I 呼出 C a p i が処理対象として指定するサービス依存リソース R S が、ステップ S 5 1 4 で検出された現在実行中の実行形式 D E に対応するサービスに対応するか否かが判断される。実行中の実行形式 D E に対応するサービスに対応する場合は、Y e s と判断されて処理は次のステップ S 5 1 8 に進む。

ステップ S 5 1 8 において、サービス依存リソース管理器 1 3 6 によって、サービス依存リソースに対するサービス依存 A P I の処理が行なわれる。そして、本ルーチンにおける処理は終了される。

一方、上述のステップ S 5 1 2 において、N o、つまりサービス依存 A P I ではない（すなわち一般 A P I である）と判断される場合、処理はステップ S 5 2 0 に進む。

ステップ S 5 2 0 において、一般リソース管理器 1 3 5 によって、一般リソースに対して処理が行われる。そして、本ルーチンにおける処理は終了される。

さらに、上述のステップ S 5 1 6 において N o、つまり実行中の実行形式 D E に対応するサービスに対応しない場合、処理はステップ S 5 2 1 に進む。

ステップ S 5 2 1 において、エラー処理が行われた後に、本ルーチンが終了される。このように、実行中の実行形式 D E が対応しているサービス（S 5 1 2 で Y e s）であっても、操作不可能（S 5 1 6 で N o）なサービス依存リソースに対する A P I 処理を許可しないように設定している。

言い換えれば、本実施の形態においては、実行中の実行形式 D E に対応付けられたサービスに関するサービス依存リソースのみが操作可能と設定される。このため、実行形式 D E によって如何なる A P I 呼び出しを発行させても、サービス依存リソース管理器 1 3 6 は他のサービスに対する参照や操作を排除できる。つまり、他のサービスに対する参照や操作を行うようなサービスの実行形式 D E が入力

されても、ステップ S 5 1 6 を経てステップ S 5 2 1 でエラー処理が行われて、そのような参照や操作を防止すると共に、そのような要求を検出できる。

上述のように、本実施の形態においては、ステップ S 5 1 2 において、リソースセクタ 1 3 4 によって、サービス依存リソースに対する処理はサービス依存 A P I でのみ操作できる様に規制されている。これについて、図 6 に示すサービス安全拡張プラットフォーム S E P 1 の端末 1 3 0 のソフトウェア階層を参照して説明する。

図 6 に示すように、サービス安全拡張プラットフォーム S E P 1 の端末 1 3 0 をソフトウェア構成から見れば、最下層に基本ソフトウェアである O S により実現される O S 層が存在する。そして、O S 層の直上に、それぞれ一般リソース管理者 1 3 5、サービス依存リソース管理者 1 3 6、および識別化実行形式格納器 1 3 2 を機能させる一般リソース管理ソフトウェア、サービス依存リソース管理ソフトウェア、および識別化実行形式格納ソフトウェアを有する。そして、一般リソース管理ソフトウェアおよびサービス依存リソース管理ソフトウェアの直上にはリソースセクタ 1 3 4 を機能させるリソースセレクトソフトウェアを有する。これらの、一般リソース管理ソフトウェア、サービス依存リソース管理ソフトウェア、リソースセレクトソフトウェア、および識別化実行形式格納ソフトウェアは、ミドルウェア層を構成する。

ミドルウェア層のリソースセレクト層の直上には、エグゼキュータ 1 3 3 を機能させるエグゼキュートソフトウェア

ア（エグゼキュータ）が位置し、識別化実行形式格納ソフトウェアの直上にはダウンローダ 1 3 1 を機能させるダウンロードソフトウェア（ダウンローダ）が位置している。そして、これらのエグゼキュートソフトウェアとダウンロードソフトウェアは共に、アプリケーション層を構成している。

そして、アプリケーション層のエグゼキュートソフトウェアの直上には、各サービス S 1 ～ S n を実行するサービス S 1 ～ S n 実行形式が位置して、コンテンツ層を構成している。

このように、図 6 に示されるソフトウェア構成から明らかなように、サービス安全拡張プラットフォーム S E P 1 においては、サービスの実行形式 D E はリソースセクタ経由の A P I 呼び出したリソースを参照、あるいは操作したりできない。よって、サービス依存リソースに対しては、サービス依存 A P I を呼び出すことが必須であるので、サービス依存リソース管理者 1 3 6 を経由しなければ参照したり操作したりできない。

次に、図 7 に示す、サービス安全拡張プラットフォーム S E P 1 によってユーザに提示される画面例を参照して、新たに追加されたサービスを実際に利用する状態に変更する際の動作について簡単に説明する。図 7 には、実行形式 D E により表示される新規サービス利用開始の是非をユーザに問い合わせる画面の一例が示されている。

画面 S M は、新たなサービスである「マイ・ニュース・サービス」を実現する実行形式 D E をエグゼキュータ 1 3

3 で実行することで提示される画面の 1 例である。画面 S M 上にはサービスの利用を宣言するボタン B Y と、利用しないことを宣言するボタン B N が配置されている。ユーザは、入力デバイス进行操作してボタン B Y を選択するとこのサービスの利用が開始される。

ここでボタン B Y が選択された場合の動作について説明する。ボタン B Y には、サービスの利用を開始を宣言するサービス依存 A P I を起動する様に実行形式 D E 中にプログラミングされている。このためボタン B Y が選択されるとリソースセクタ 1 3 4 を経由してサービス依存 A P I がサービス依存リソース管理器 1 3 6 に届き、サービス依存リソース管理テーブル T r s に記載されているサービス依存リソース R S に該当するサービスの利用状態の欄の値を「利用」に書き換える。

なお、図 7 に示す例では、サービスの単純な利用開始について述べたが、サービス提供部に対するサービス利用契約の締結やユーザ登録、あるいは蓄積型のデータ放送における視聴前の事前の自動蓄積処理開始なども同様に実施可能である。

このように、サービスの使用／未使用の状態遷移など、サービスに関する操作をサービス自身の実行形式 D E によってユーザとの対話を行うことが可能となる。さらに、如何なる悪意を持った実行形式 D E を他のサービスが実装された場合であっても、サービスやプラットフォームに対する、誤動作やハングアップに代表される、あらゆる悪影響を排除できる。

上述のように、本発明の第 1 の実施の形態にかかるサービス安全拡張プラットフォーム S E P 1 においては、実行形式 D E に不正な処理を引き起こすコードが含まれていないことを実行形式 D E の配布元の身元の確からしさから類推するような不確実な安全性ではなく、どのような実行形式 D E を想定しても他のサービスやプラットフォーム自体に対する不正な処理の影響を引き起こさない、完全な安全性を、サービスの拡張性を保ったままで確保できる。

(第 2 の実施の形態)

以下に、図 8、図 9、および図 11 を参照して、本発明の第 2 の実施の形態にかかるサービス安全拡張プラットフォームについて説明する。図 8 に示すように、本実施の形態にかかるサービス安全拡張プラットフォーム S E P 2 は、図 1 に示すサービス安全拡張プラットフォーム S E P 1 におけるサービス提供源 110 がサービス提供源 210 に変更され、端末 130 が端末 230 に変更されている。

サービス提供源 210 は、サービス提供源 110 におけるサービス識別設定器 112 がメタサービス指定器 212 に交換されている。端末 230 は端末 130 におけるエグゼキュータ 133 がエグゼキュータ 233 に交換されると共に、リソースセレクタ 134 とサービス依存リソース管理者 136 との間にメタサービス判定器 234 が新たに設けられている。

端末 230 は、第 1 の実施の形態における端末 130 のダウンローダ 131 がダウンローダ 231 に置き換えられ、識別化実行形式格納器 132 が識別化実行形式格納器 2



3 2 に置き換えられ、エグゼキュータ 1 3 3 がエグゼキュータ 2 3 3 に置き換えられると共に、リソースセクタ 1 3 4 とサービス依存リソース管理者 1 3 6 との間にメタサービス判定器 2 3 4 が新たに設けられている。

サービス提供源 2 1 0 において、メタサービス指定器 2 1 2 は、上述のサービス識別設定器 1 1 2 にメタサービス指定情報生成機能が付与されている。つまり、メタサービス指定器 2 1 2 は、サービス識別設定器 1 1 2 と同様に、実行形式 D E の属性等の副次的な事項を表す付随情報 I S を生成すると共に、メタサービスに対応付ける実行形式 D E を指定するメタサービス指定情報 I S m を生成する。

メタサービスとは、上述のサービスの一種であるが、実際にユーザの利用するサービスの利用開始／終了などのサービスの機能／諸元の変更や新規サービスの追加といったサービスの拡張を行うことができる唯一のサービスであり、サービスの拡張を目的として存在するものである。この意味において、本明細書においては上述の第 1 の実施の形態にかかるサービス S o と、本実施の形態におけるメタサービス S m e t a を区別して説明する。

付随情報 I S が実行形式 D E に付与されて識別化実行形式 D E i が生成され、メタサービス指定情報 I S m が実行形式 D E に付与されてメタサービス実行形式 D E m が生成される。つまり、メタサービス指定器 2 1 2 からは、識別化実行形式 D E i とメタサービス実行形式 D E m が混在して出力される。そして、これらの識別化実行形式 D E i とメタサービス実行形式 D E m は、送出器 1 1 3 およびデリ

バリシステム 1 2 0 を介して端末 2 3 0 に入力される。そして、端末 2 3 0 のダウンロード 2 3 1 によって、識別化実行形式 D E i とメタサービス実行形式 D E m は識別化実行形式格納器 2 3 2 に出力される。そして、ダウンロード 2 3 1 は識別化実行形式 D E i から付随情報 I S を抽出し、メタサービス実行形式 D E m からメタサービス指定情報 I S m を抽出して、それぞれをサービス依存リソース管理器 2 3 6 に出力する。

図 9 に、メタサービス指定器 2 1 2 から出力された識別化実行形式 D E i とメタサービス実行形式 D E m が識別化実行形式格納器 2 3 2 に格納される様子を示す。同図に示す例においては、図 2 に示した第 1 の実施の形態におけるのと同様に、サービス S 1、サービス S 2 乃至サービス S n (n は任意の自然数) に対応する識別化実行形式 D E i (S 1)、D E i (S 2)、および D E i (S n) が例示されている。ただし、本図においては、サービス S 3 に対応する識別化実行形式 D E i (S 3) の位置に、初めてのメタサービス実行形式 D E m (1) が表示されている。メタサービス実行形式 D E m (1) は、実行形式 D E に、それがメタサービスであることを示すメタサービス指定情報 I S m (1) が付与されている。

メタサービス実行形式 D E m (1) は、それがメタサービス m 1 に対応するサービス識別情報 E s (S m 1) と副次情報  $\alpha$  m 1 から成るメタサービス指定情報 I S m と実行形式 D E を含む。具体的には、サービス識別情報 E s (S m 1) が実行形式 D E とメタサービス 1 とを対応付けてい

る。このようにサービス識別情報  $E_s$  ( $S_{m1}$ ) の「 $S_{m1}$ 」がメタサービス 1 を規定する点を除けば、メタサービス指定情報  $I_{S_m}$  も基本的には付随情報  $I_S$  と同じものである。

つまり、メタサービス  $S_m$  は、上述のように、サービス  $S_n$  の 1 つであるので、メタサービス実行形式  $DE_m$  は、識別化実行形式  $DE_i$  ( $S_m$ ) と表すこともできる。よって、メタサービス指定情報  $I_{S_m}$  も付随情報  $I_S$  と総称してもよいが、本明細書においては、本実施の形態における特徴をわかりやすくするために、メタサービス指定情報  $I_{S_m}$  を付随情報  $I_S$  と区別して説明する。さらに、メタサービスに対応するサービス識別情報  $E_s$  を、メタサービス識別情報  $E_{s_m}$  と称して、実行形式  $DE$  に対応するサービス識別情報  $E_s$  と区別して説明する。

識別化実行形式格納器 232 の動作は、上述の第 1 の実施の形態にかかる識別化実行形式格納器 132 と基本的に同じである。ただし、識別化実行形式格納器 232 は、識別化実行形式  $DE_i$  とメタサービス実行形式  $DE_m$  とを格納し、要求に応じて、識別化実行形式  $DE_i$  あるいはメタサービス実行形式  $DE_m$  をエグゼキュータ 233 に出力する。

エグゼキュータ 233 は、識別化実行形式  $DE_i$  からサービス識別情報  $E_s$  を抽出し、メタサービス実行形式  $DE_m$  からサービス識別情報  $E_{s_m}$  を抽出して、それぞれをメタサービス判定器 234 に出力する。

メタサービス判定器 234 は、リソースセレクタ 134

から入力される A P I 呼出 C a p i と、エグゼキュータ 1 3 3 から入力されるサービス識別情報 E s およびサービス識別情報 E s m に基づいて、実行中の実行形式 D E がメタサービスに対応付けられている場合にのみ、A P I 呼出 C a p i をサービス依存リソース管理者 1 3 6 に出力する。

サービス依存リソース管理者 1 3 6 は、サービス依存リソース R S を格納すると共にサービス依存リソース管理テーブル T r s によってサービス依存リソース R S を管理する。なお、サービス依存リソース管理者 1 3 6 に格納されるサービス依存リソース R S およびそのサービス依存リソース管理テーブル T r s は、図 3 を参照して説明した第 1 の実施の形態におけるものと同じでよい。

そして、サービス依存リソース管理者 1 3 6 は、メタサービス判定器 2 3 4 を経由して入力される A P I 呼出 C a p i に応答して、サービス依存リソース管理者 1 3 6 に格納されているサービス依存リソース R S への参照や操作などを行う。また、ダウンローダ 1 3 1 から供給される付随情報 I S に基づいて、実行形式 D E のそれぞれが対応するサービスの内容を認識する。

新たなサービスの追加は、サービス依存リソース管理者 1 3 6 が、ダウンローダ 2 3 1 から入力される付随情報 I S およびメタサービス指定情報 I S m に基づいて、サービス依存リソース管理テーブル T r s に登録されていない新規のサービスの存在を検出した時点で、そのサービスに対する識別行を追加して、内容を記述する。つまり、サービス依存リソース管理テーブル T r s には、列 C 1 にはメタ

サービスの内容が書き込まれるが、サービス識別行追加の手順だけに注目すれば、第1の実施の形態における処理内容と同様である。

次に、図10に示すフローチャートを参照して、端末230によるサービス実行ルーチンの動作について説明する。具体的には、端末230において、エグゼキュータ233が呼び出すAPIに対する実行形式DEが実行されることによって、サービス実行が実現される。つまり、エグゼキュータ133が識別化実行形式格納器132から入力される識別化実行形式DE<sub>i</sub>およびメタサービス実行形式DE<sub>m</sub>に含まれる実行形式DEを実行させるために、API呼出Cap<sub>i</sub>を発行した時点で、本ルーチンにおける処理が開始される。

なお、図10に示すフローチャートは、上述の図5に示すフローチャートにおいて、「実行中の実行形式DEのサービス識別情報E<sub>s</sub>を実行エンジンから取得」ステップS514が「実行中の実行形式DEのサービス識別情報E<sub>s</sub>またはサービス識別情報E<sub>s m</sub>を取得」ステップS1014に交換され、「実行中のサービスに対する操作かを判定」ステップS516が「実行中のメタサービスに対する操作かを判定」ステップS1016に交換されている点を除いては同様に構成されている。以下、本実施の形態に固有の動作に重点をおいて説明する。

よって、ステップS512において、リソースセクタ134によって、エグゼキュータ133から発行されたAPI呼出Cap<sub>i</sub>に基づいて、呼び出されたAPIがサー

ビス依存 A P I であると判断されて、処理はステップ S 1 0 1 4 に進む。

ステップ S 1 0 1 4 において、メタサービス判定器 2 3 4 によって、エグゼキュータ 2 3 3 から現在実行中の実行形式 D E に対するサービス識別情報 E s あるいはメタサービス識別情報 E s m が読み出される。そして、処理は次のステップ S 1 0 0 6 に進む。

ステップ S 1 0 1 6 において、実行中の実行形式 D E がメタサービスに対応付けられているか否かが判断される。N o の場合、上述のエラー処理ステップ S 5 2 1 を経て、本ルーチンが終了される。一方、実行中の実行形式 D E がメタサービスに対応付けられている場合は、Y e s と判断されて、処理は上述の「サービス依存リソースに対して処理」ステップ S 5 1 8 を経て本ルーチンが終了される。

なお、ステップ S 5 1 8 においては、実行中のメタサービスの実行形式 D E であれば（ステップ S 1 0 1 6 で Y e s ）、全てのサービスに対するサービス依存リソースに対する処理が実行される。

上述のように、第 2 の実施の形態においては、実行形式 D E がメタサービスであれば、全てのサービスに対するサービス依存 A P I が実行できる。従ってメタサービスの実行形式 D E の画面上で、サービス提供部から提供可能なサービス一覧を表示し、一覧表示上で、サービスの機能／諸元の変更や新規サービスの追加に代表されるサービスの拡張が達成される。

（第 3 の実施の形態）



次に、図 1 1、図 1 2、および図 1 3 を参照して、本発明の第 3 の実施の形態にかかるサービス安全拡張プラットフォームについて詳細に説明する。図 1 1 に示すように、本実施の形態にかかるサービス安全拡張プラットフォーム S E P 3 は、図 1 に示すサービス安全拡張プラットフォーム S E P 1 におけるサービス提供源 1 1 0 がサービス提供源 3 1 0 に変更され、端末 1 3 0 が端末 3 3 0 に変更されている。

サービス提供源 3 1 0 は、サービス提供源 1 1 0 におけるサービス識別設定器 1 1 2 がサービス識別設定器 3 1 2 に交換されると共に、サービス識別親子管理者 3 1 4 が新たに設けられている。サービス識別親子管理者 3 1 4 は、個々のサービスに親子関係がある場合に、そのような関係を識別情報 E s の間の親子関係として管理する。さらに、管理している親子関係を示すサービス親子指定情報 I S h をサービス識別設定器 3 1 2 に出力する。

端末 3 3 0 は、端末 1 3 0 におけるダウンロード 1 3 1 がダウンロード 3 3 1 に交換され、識別化実行形式格納器 1 3 2 が識別化実行形式格納器 3 3 2 に交換され、エグゼキュータ 1 3 3 がエグゼキュータ 2 3 3 に交換され、親子判定器 3 3 4 がリソースセレクタ 1 3 4 とサービス依存リソース管理者 1 3 6 の間に新たに設けられている。

サービス提供源 3 1 0 において、サービス識別親子管理者 3 1 4 が管理するサービスの親子関係とは、親サービスが子サービスのサービス依存リソース R s を操作可能とする関係と定義される。なお、必要に応じて、親サービス S

p、子サービス S c、親サービスのサービス依存リソース R S c を称して、互いに識別するのとする。例えば、「音楽コンテンツ配信サービス」といったサービスのカテゴリに対応する親サービスに対して、「松下ミュージック配信サービス」や、「テイチクミュージック配信サービス」、および「イーピーチャンネル音楽サービス」に代表される個々のサービスの種類に対応する子サービスがある。

サービス識別設定器 3 1 2 は、上述のサービス識別設定器 1 1 2 に親子サービス識別情報生成機能が付与されている。つまり、サービス識別設定器 3 1 2 は、サービス識別設定器 1 1 2 と同様に、実行形式 D E の属性等の副次的な事項を表す付随情報 I S を生成する。さらに、サービス識別設定器 3 1 2 は、サービス識別親子管理者 3 1 4 から供給されるサービス親子情報 I h に基づいて、サービス識別設定器 3 1 2 は、サービス親子情報 E s c を生成する。例えば、サービス親子情報 I h に基づいて、サービス親子指定情報 I S h ( S 1 - 1 ) が生成される。サービス親子情報 E s c ( S 1 - 1 ) は、このサービスが親サービス S 1 の子サービスであることを定義する。

付随情報 I S が実行形式 D E に付与されて識別化実行形式 D E i が生成され、サービス親子指定情報 I S h が実行形式 D E に付与されて親子識別化実行形式 D E c が生成される。つまり、サービス識別設定器 3 1 2 からは、識別化実行形式 D E i と親子識別化実行形式 D E c が混在して出力される。そして、これらの識別化実行形式 D E i と親子識別化実行形式 D E c は、送出器 1 1 3 およびデリバリシ

ステム 1 2 0 を介して端末 3 3 0 に入力される。そして、端末 3 3 0 のダウンロード 3 3 1 によって、識別化実行形式 D E i と親子識別化実行形式 D E c は識別化実行形式格納器 3 3 2 に出力される。さらに、ダウンロード 3 3 1 は、識別化実行形式 D E i から付随情報 I S を抽出し、親子識別化実行形式 D E c からサービス親子指定情報 I S h を、それぞれをサービス依存リソース管理者 1 3 6 に出力する。

図 1 2 に、サービス識別設定器 3 1 2 から出力された識別化実行形式 D E i と親子識別化実行形式 D E c が識別化実行形式格納器 3 3 2 に格納される様子を示す。同図に示す例においては、図 2 に示した第 1 の実施の形態におけるのと同様に、サービス S 1、サービス S 2 乃至サービス S n (n は任意の自然数) に対応する識別化実行形式 D E i (S 1)、D E i (S 2)、および D E i (S n) が例示されている。ただし、本図においては、サービス S 3 に対応する識別化実行形式 D E i (S 3) の位置に、初めての親子識別化実行形式 D E c (1) が表示されている。親子識別化実行形式 D E c (1) は、実行形式 D E に、それが子サービスであることを示すサービス親子指定情報 I S h (1) が付与されている。

親子識別化実行形式 D E c (1) は、それが親サービス S p 1 に対応するサービス識別情報 E s (S c 1) と副次情報  $\alpha$  c 1 から成るサービス親子指定情報 I S h (1) と実行形式 D E を含む。具体的には、親子サービス識別情報 E s c が実行形式 D E を親サービス S p 1 に対する子サー

ビス S c 1 として対応付けている。このようにサービス識別情報 E s ( S p 1 ) の「 S p 1 」が子サービス 1 を規定する点を除けば、サービス親子指定情報 I S h も基本的には付随情報 I S と同じものである。

つまり、子サービス S c は、上述のように、サービス S n の 1 つであるので、親子識別化実行形式 D E c は、識別化実行形式 D E i ( S c ) と表すこともできる。よって、サービス親子指定情報 I S h も付随情報 I S と総称してもよいが、本明細書においては、本実施の形態における特徴をわかりやすくするために、サービス親子指定情報 I S h を付随情報 I S と区別して説明する。さらに、親サービス S p に対する子サービス S c を規定するサービス識別情報 E s を親子サービス識別情報 E s c と称して、実行形式 D E に対応するサービス識別情報 E s と区別して説明する。

識別化実行形式格納器 3 3 2 の動作は、上述の第 1 の実施の形態にかかる識別化実行形式格納器 1 3 2 と基本的に同じである。ただし、識別化実行形式格納器 3 3 2 は、識別化実行形式 D E i と親子識別化実行形式 D E c とを格納し、要求に応じて、識別化実行形式 D E i あるいは親子識別化実行形式 D E c をエグゼキュータ 3 3 3 に出力する。

エグゼキュータ 3 3 3 は、識別化実行形式 D E i からサービス識別情報 E s を抽出し、親子識別化実行形式 D E c から親子サービス識別情報 E s c を抽出して、それぞれを親子判定器 3 3 4 に出力する。

親子判定器 3 3 4 は、リソースセレクタ 1 3 4 から入力される A P I 呼出 C a p i と、エグゼキュータ 3 3 3 から

入力されるサービス識別情報 E s および親子サービス識別情報 E s c 基づいて、実行中の実行形式 D E が A P I 呼出 C a p i の処理対象のサービス依存リソースのサービスの先祖（親、または親の親、または親の親の親．．．）である場合にのみ、A P I 呼出 C a p i をサービス依存リソース管理器 1 3 6 に出力する。

つまり、親子判定器 3 3 4 は、実行中の実行形式 D E の親子サービス識別情報 E s c をエグゼキュータ 3 3 3 から得るとともに、エグゼキュータ 3 3 3 から発行される A P I 呼出 C a p i をリソースセレクタ 1 3 4 を介して得る。さらに、ダウンローダ 3 3 1 からサービス間の親子関係を判定するための情報であるサービス親子指定情報 I S h を得る。そして、親子サービス識別情報 E s c が示すサービスが、サービス親子指定情報 I S h の示すサービスの先祖であるかを判断する。そして、先祖であると判断した場合には、サービス依存リソース管理器 1 3 6 に対してサービス依存 A P I の呼び出し要求（A P I 呼出 C a p i）を伝え、先祖でないと判断した場合には伝えない。

次に、図 1 3 に示すフローチャートを参照して、端末 3 3 0 におけるサービス実行ルーチンの動作について説明する。具体的には、端末 3 3 0 において、エグゼキュータ 3 3 3 が呼び出す A P I に対する実行形式 D E および親子識別化実行形式 D E c に含まれる実行形式 D E を実行させるために、A P I 呼出 C a p i を発行した時点で、本ルーチンにおける処理が開始される。

なお、図 1 3 に示すフローチャートは、上述の図 5 に示



すフローチャートにおいて、「実行中の実行形式 D E のサービス識別情報 E s を実行エンジンから取得」ステップ S 5 1 4 が「実行中の実行形式 D E のサービス識別情報 E s または親子サービス識別情報 E s c を取得」ステップ S 1 3 1 4 に交換され、「実行中のサービスに対する操作かを判定」ステップ S 5 1 6 が「実行中のメタサービスに対する操作かを判定」ステップ S 1 3 1 6 に交換されている点を除いては同様に構成されている。以下、本実施の形態に固有の動作に重点をおいて説明する。

よって、ステップ S 5 1 2 において、リソースセクタ 1 3 4 によって、エグゼキュータ 3 3 3 から発行された A P I 呼出 C a p i に基づいて、呼び出された A P I がサービス依存 A P I であると判断されて、処理はステップ S 1 3 0 4 に進む。

ステップ S 1 3 1 4 において、親子判定器 3 3 4 によって、エグゼキュータ 3 3 3 から現在実行中の実行形式 D E に対するサービス識別情報 E s あるいは親子サービス識別情報 E s c が読み出される。そして、処理は次のステップ S 1 3 0 6 に進む。

ステップ S 1 3 1 6 において、親子判定器 3 3 4 によって、サービス依存 A P I 呼出 C a p i の処理対象のサービス依存リソースに対応するサービスに対して、実行中の実行形式 D E の対応するサービスが先祖のサービスであるかが判断される。N o の場合、上述のエラー処理ステップ S 5 2 1 を経て、本ルーチンが終了される。一方、実行中の実行形式 D E が実行中のサービスが先祖である場合は、



、Y e s と判断されて、処理は上述の「サービス依存リソースに対して処理」ステップ S 5 1 8 を経て本ルーチンが終了される。

なお、ステップ S 5 1 8 においては、サービス依存リソース管理者 1 3 6 によって、サービス依存 A P I 呼出 C a p i に対応するサービス依存リソースに対する処理が実行される。すなわち、現在実行している実行形式 D E のサービスの子孫（子、または子の子、または子の子の子、．．．）のサービス依存リソースに対して処理が可能になるように管理される。

#### （第 4 の実施の形態）

次に、図 1 4 および図 1 5 、図 1 6 、および図 1 7 を参照して、本発明の第 4 の実施の形態にかかるサービス安全拡張プラットフォームについて説明する。図 8 に示すように、本実施の形態の形態にかかるサービス安全拡張プラットフォーム S E P 4 は、図 1 に示すサービス安全拡張プラットフォーム S E P 1 におけるサービス提供源 1 1 0 がサービス提供源 4 1 0 に変更され、端末 1 3 0 が端末 4 3 0 に変更されている。

サービス提供源 4 1 0 は、サービス提供源 1 1 0 における実行形式提供器 1 1 1 がコンテンツ提供器 4 1 1 に交換され、サービス識別設定器 1 1 2 が制御コンテンツ指定器 4 1 2 に交換されている。端末 4 3 0 は、端末 1 3 0 におけるダウンローダ 1 3 1 がコンテンツダウンローダ 4 3 1 に交換され、識別化実行形式格納器 1 3 2 がコンテンツ格納器 4 3 2 に交換され、エグゼキュータ 1 3 3 がエグゼキ

ユーザ 4 3 3 に交換されると共に、リソースセクタ 1 3 4 とサービス依存リソース管理者 1 3 6 との間に制御コンテンツ判定器 4 3 4 が新たに設けられている。

コンテンツ提供器 4 1 1 は、サービスを実現する実行形式 D E と実行形式 D E が解釈しユーザに提示する目的で製作されたデータであるコンテンツ D C を格納し、必要に応じて出力する。実行形式 D E もコンテンツ D C の満たすべき要件を備えている。例えば、共に、実行形式 D E もコンテンツ D C も共に、逐次動作手順をプログラミングするスクリプト記述を含めることが可能なマークアップ言語で記述する。スクリプトの言語として J a v a ( R ) 系の J a v a S c r i p t や E C M A S c r i p t などの言語でよい。また、マークアップ言語としては、H T M L 言語や B M L 言語でよい。

制御コンテンツ指定器 4 1 2 は、コンテンツ提供器 4 1 1 が出力するコンテンツ D C のうち、制御コンテンツに対して制御コンテンツであることを指定する付随情報 I S C を付与して制御コンテンツ D C c として出力する。制御コンテンツ D C c の定義は、サービスに対応付けられた実行形式 D E として製作されたコンテンツ D C である。

コンテンツダウンローダ 4 3 1 は、デリバリシステム 1 2 0 から伝送されてくる、コンテンツ D C と制御コンテンツ D C c とを受信し、コンテンツ格納器 4 3 2 に受信した情報を書き込む。

デリバリシステム 1 2 0 からは実行形式 D E もコンテンツ D C の一種類として伝送されるため、コンテンツダウン

ローダ 4 3 1 は、コンテンツ D C を格納する機能を有していればよい。

コンテンツダウンローダ 4 3 1 は、サービス依存リソース管理者 1 3 6 がサービス依存リソースとして保持する、サービスのコンテンツを自動ダウンロードすべきかを表す情報に基づきコンテンツの格納を制御する。

すなわち、コンテンツダウンローダ 4 3 1 は、注目するサービスに対し、自動ダウンロードすべき場合には、サービスに関するコンテンツ D C を制御コンテンツ D C c も含め全て格納する。一方、自動ダウンロードすべきでない場合には、制御コンテンツ D C c のみを格納する。

コンテンツ格納器 4 3 2 は、コンテンツ D C を格納する。第 1 の実施の形態の識別化実行形式格納器 1 3 2 が格納する情報が実行形式 D E であるのに対し、コンテンツ格納器 4 3 2 はコンテンツ D C を格納する。なお、コンテンツ D C には制御コンテンツ D C c が含まれる。

図 1 5 に、制御コンテンツ指定器 4 1 2 から出力されたコンテンツ D C と付随情報 I S c がコンテンツ格納器 4 3 2 に格納される様子を示す。同図に示す例においては、サービス S 1 に対して、制御コンテンツ D C c である C ( S 1 , D E ) 、コンテンツ D C である C ( S 1 , 1 ) 、 C ( S 1 , 2 ) 、および C ( S 1 , 3 ) がコンテンツ格納器 4 3 2 に保持されている。制御コンテンツ D C c である C ( S 1 , E ) をエグゼキュータ 4 3 3 で実行させることで、コンテンツ D C である C ( S 1 , 1 ) 、 C ( S 1 , 2 ) 、および C ( S 1 , 3 ) を読み込んでユーザに提示する。す

なわち制御コンテンツDCcであるC(S1, DE)は、コンテンツDCであるC(S1, 1)、C(S1, 2)、C(S1, 3)に対するいわゆるブラウザと同等の動作を行う。

制御コンテンツ判定器434は、エグゼキュータ433の出力から、実行しているコンテンツDCが制御コンテンツDCcであるかを判定する。そして、制御コンテンツDCcである場合のみに、リソースセクタ134からのサービス依存APIの要求をサービス依存リソース管理者136に伝える。

図16に、本実施の形態において、サービス依存リソース管理者136が格納するサービス依存リソーステーブルTrs4の一例である。サービス依存リソーステーブルTrs4は、図3に示した、第1の実施の形態にかかるサービス依存リソース管理テーブルTrsと類似しているが、列C2には、コンテンツダウンローダ431がコンテンツDCを自動ダウンロードすべきかについての情報がサービス毎に保持されている。

次に、図17に示すフローチャートを参照して、端末430における制御コンテンツDCcの実行ルーチンの動作について説明する。具体的には、端末430において、エグゼキュータ433が呼び出すAPIに対する制御コンテンツDCcを実行するために、API呼出Cap iを発行した時点で本ルーチンにおける処理が開始される。

なお、図17に示すフローチャートは、上述の図5に示すフローチャートにおいて、実行中の実行形式DEのサー

ビス識別情報 E s を実行エンジンから取得」ステップ S 5 1 4 が「実行中のコンテンツ D C のサービス識別情報 E s を取得」ステップ S 1 7 1 4 に交換され、「実行中のサービスに対する操作かを判定」ステップ S 5 1 6 が「実行中のサービスに対する制御コンテンツ操作かを判定」ステップ S 1 7 1 6 に交換されている点を除いては同様に構成されている。以下、本実施の形態に固有の動作に重点をおいて説明する。

ステップ S 5 1 2 において、リソースセクタ 1 3 4 によって、エグゼキュータ 3 3 3 から発行された A P I 呼出 C a p i に基づいて、呼び出された A P I がサービス依存 A P I であると判断されて、処理はステップ S 1 7 0 4 に進む。

ステップ S 1 7 1 4 において、制御コンテンツ判定器 4 3 4 によって、エグゼキュータ 4 3 3 から現在実行中のコンテンツ D C に対するサービス識別情報 E s が読み出される。そして、処理は次のステップ S 1 7 1 6 に進む。

ステップ S 1 7 1 6 において、制御コンテンツ判定器 4 3 4 によって、サービス依存 A P I 呼出 C a p i の処理対象のサービス依存リソースに対応するサービスに対して、実行中のコンテンツ D C がコンテンツ D C の対応するサービスに対応する制御コンテンツであるか否かが判断される。N o の場合、上述のエラー処理ステップ S 5 2 1 を経て、本ルーチンが終了される。一方、実行中のコンテンツ D C が実行中のサービスが先祖である場合は、Y e s と判断されて、処理は上述の「サービス依存リソースに対して処

理」ステップ S 5 1 8 を経て本ルーチンが終了される。

なお、ステップ S 5 1 8 においては、サービス依存リソース管理器 1 3 6 によって、サービス依存 A P I 呼出 C a p i に対応するサービス依存リソースに対する処理が実行される。

本明細書で開示したサービス安全拡張プラットフォームによれば、サービスを実現する実行形式 D E による不正アクセスの排除を、実行形式 D E に不正アクセスを引き起こす処理が含まれることを否定する必要無く達成するという、新次元の安全性を実現することができる。

詳述すれば、他のサービスの状態変更やデータの破壊や、プラットフォーム自体のシステムダウンなど、サービスの正常な享受に悪影響を及ぼす挙動を引き起こす処理である不正アクセスの排除を、実行形式 D E に不正アクセスを引き起こす処理が含まれることを実行形式 D E の配布元の確からしさから推測するまでもなく達成する。

さらには上記の性質と同時に、実行形式 D E の変更や追加によってサービスの機能／諸元の変更や新規サービスの追加といったサービスの拡張も達成する機能を実現することができる。

これにより、従来のサービスの拡張可能なプラットフォームに対して従来にない新次元の安全性と操作性とを同時に実現することができる。

#### 産業上の利用可能性

以上のように、本発明は、「サービスの拡張性」、「サ



ービス安全性の確保」および「サービス拡張操作のシームレス」という3つの特徴を実現する従来にない全く新しいサービス安全拡張プラットフォームに有用である。

## 請求の範囲

1. サービス（S）と実行形式（DE）とが対応付けられており、前記実行形式（DE）の変更や追加によって前記サービス（S）の拡張が達成されるサービス安全拡張プラットフォーム（SEP）であって、前記サービスの拡張を行うサービス依存APIを具備し、かつ前記実行形式（DE）からの前記サービスの拡張は前記サービス依存APIの呼び出しによってのみ行われることを特徴とする、サービス安全拡張プラットフォーム（SEP）。
2. 前記サービスの拡張が新規サービスの新設であることを特徴とする、請求項1に記載のサービス安全拡張プラットフォーム（SEP1）。
3. 前記サービスの拡張がサービス利用開始であることを特徴とする、請求項1に記載のサービス安全拡張プラットフォーム（SEP1）。
4. 注目する前記サービスの拡張は、注目する前記サービスに対応付けられた実行形式（DE）からの前記サービス依存APIの呼び出しによってのみ行われることを特徴とする、請求項1乃至請求項3に記載のサービス安全拡張プラットフォーム（SEP）。
5. 複数の前記サービス間に親子関係（サービス親子指定情報 ISh）が定義され、前記実行形式（DE）が要求する前記サービス依存API呼び出しが前記サービス（S）に対応付けられるサービス依存リソース（RS）を処理対象として指定した際に、前記実行形式（DE）に対応付け

られるサービス（S）が前記サービス（S）の先祖である場合にのみ、サービス依存リソース（RS）に対して処理可能であることを特徴とする、請求項1乃至請求項3に記載のサービス安全拡張プラットフォーム（SEP3）。

6. メタサービスに対応付けられた前記実行形式（DE）が前記サービス依存APIによって、前記サービス（S）の少なくとも1つが拡張可能なことを特徴とする、請求項1乃至請求項3に記載のサービス安全拡張プラットフォーム（SEP）。

7. 前記メタサービスに対応付けられた前記実行形式が前記サービス依存APIによって、前記サービス（S）の全てが拡張可能であり、前記メタサービスに対応付けられていない前記実行形式（DE）は前記サービス依存APIによって前記サービスの拡張が不可能であることを特徴とする、請求項6に記載のサービス安全拡張プラットフォーム。

8. 前記実行形式（DE）がコンテンツ（DC）として満たすべき条件を満たした制御コンテンツ（DCc）であり、前記制御コンテンツ（DC）が、前記コンテンツの少なくとも1つと共にコンテンツ（DC）として伝送され、前記コンテンツ（DC）の少なくとも1つから前記制御コンテンツ（DCc）を指定する情報（Es）が伝送され、前記制御コンテンツ（DC）によってのみサービス依存APIの処理が可能であることを特徴とする請求項1乃至請求項3に記載のサービス安全拡張プラットフォーム（SEP4）。

9. 前記サービス依存APIによって、特定の前記サービスのコンテンツ(DC)の自動的な格納が制御されることを特徴とする請求項8に記載のサービス安全拡張プラットフォーム(SEP4)。

10. 前記実行形式(DE)を少なくとも1つのサービス提供部(110、210、310、410)から送出し、前記実行形式(DE)を実行する少なくとも1つの端末(130、230、330、430)で受信することを特徴とする、請求項1乃至請求項9に記載のサービス安全拡張プラットフォーム(SEP)。

11. 請求項1乃至請求項10に記載のサービス安全拡張プラットフォーム(SEP)を実現するサービス安全拡張方法。

12. 請求項1乃至請求項10に記載のサービス安全拡張プラットフォーム実施するコンピュータプログラムを格納した記憶媒体。

**图 1**

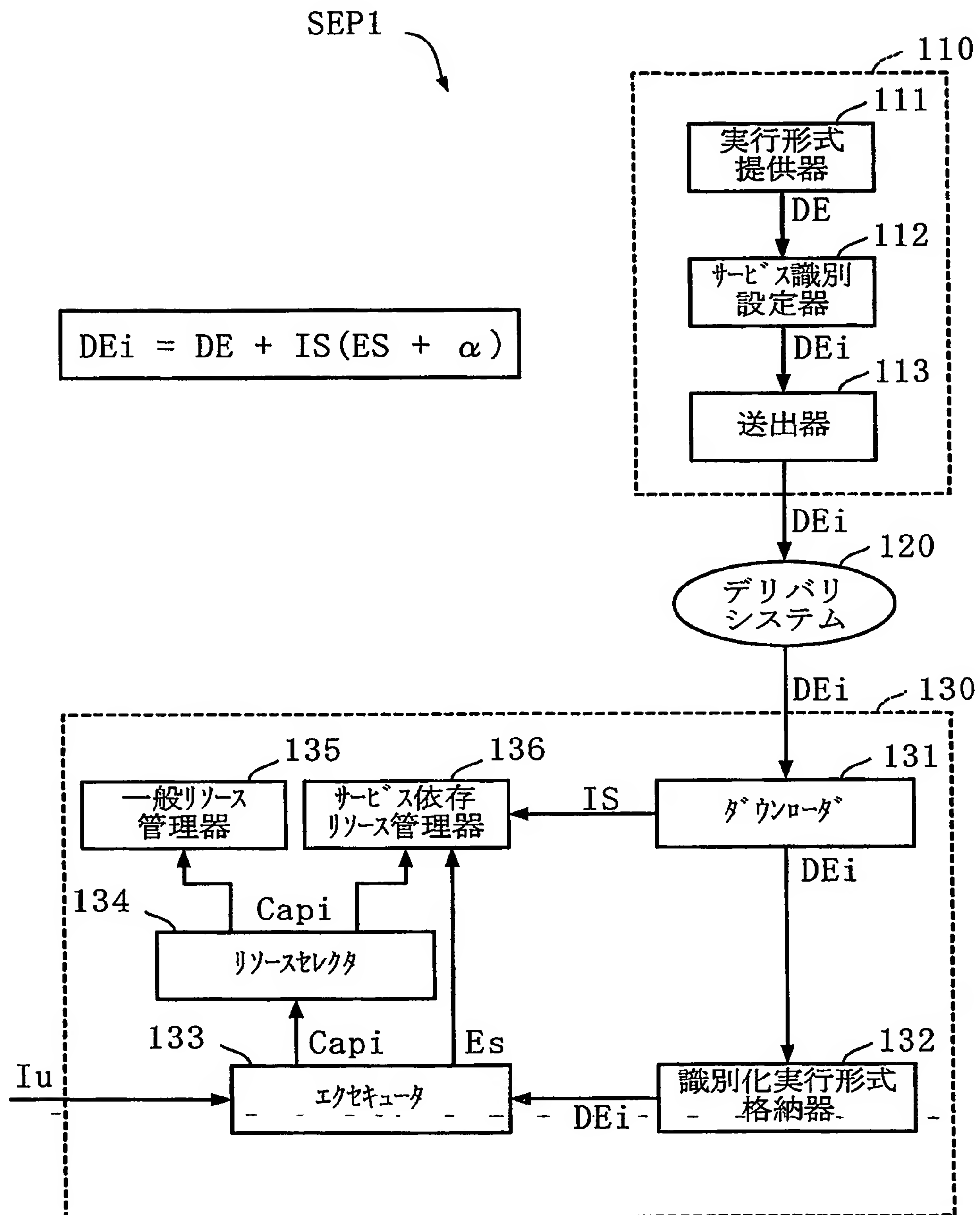


図 2

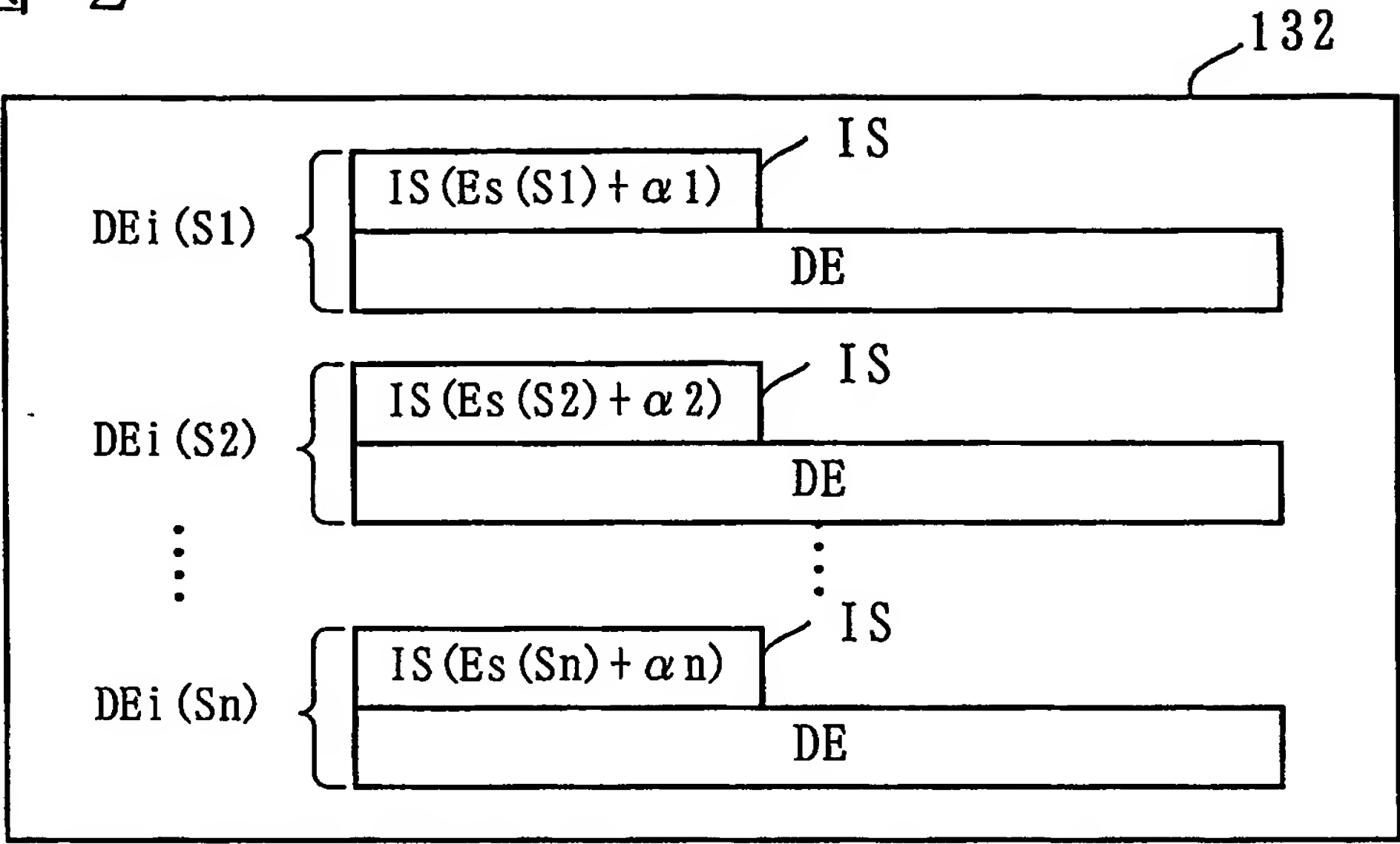


図 3

Trs

C1

C2

	サービス識別 $Es$	利用状態
L1 →	S1	未利用
L2 →	S2	利用
	⋮	⋮
Ln →	Sn	利用



図 4

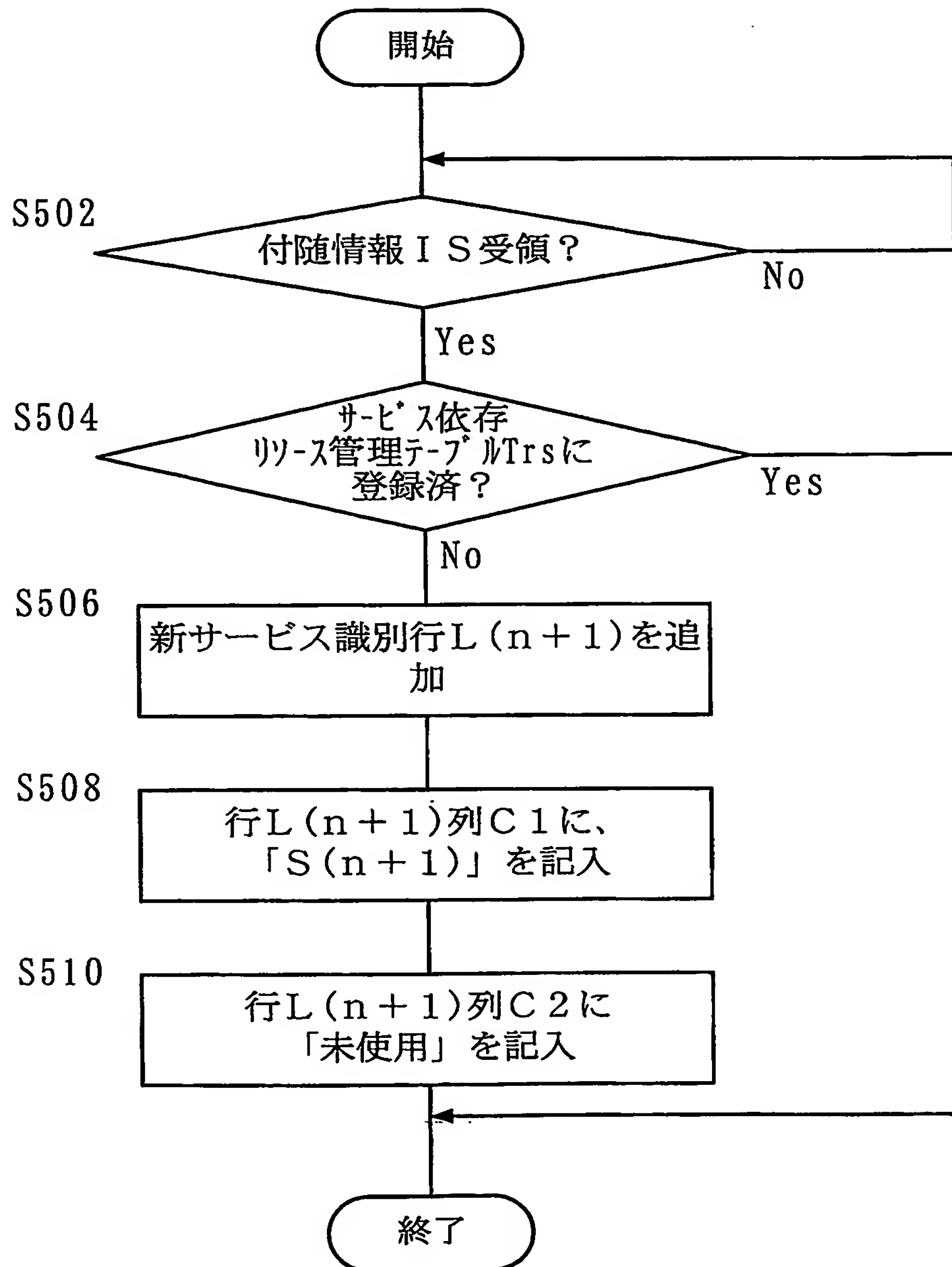


図 5

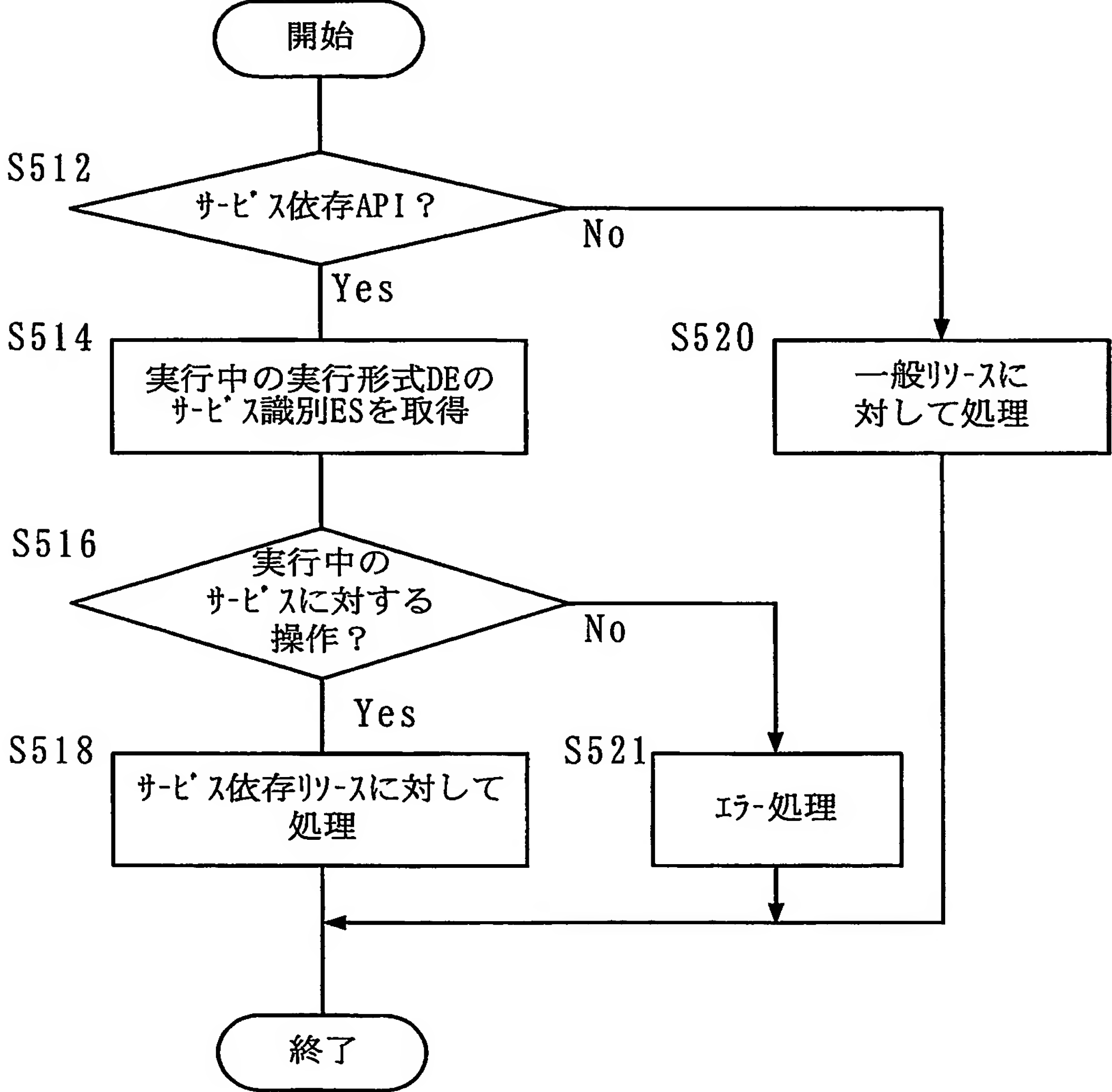


図 6

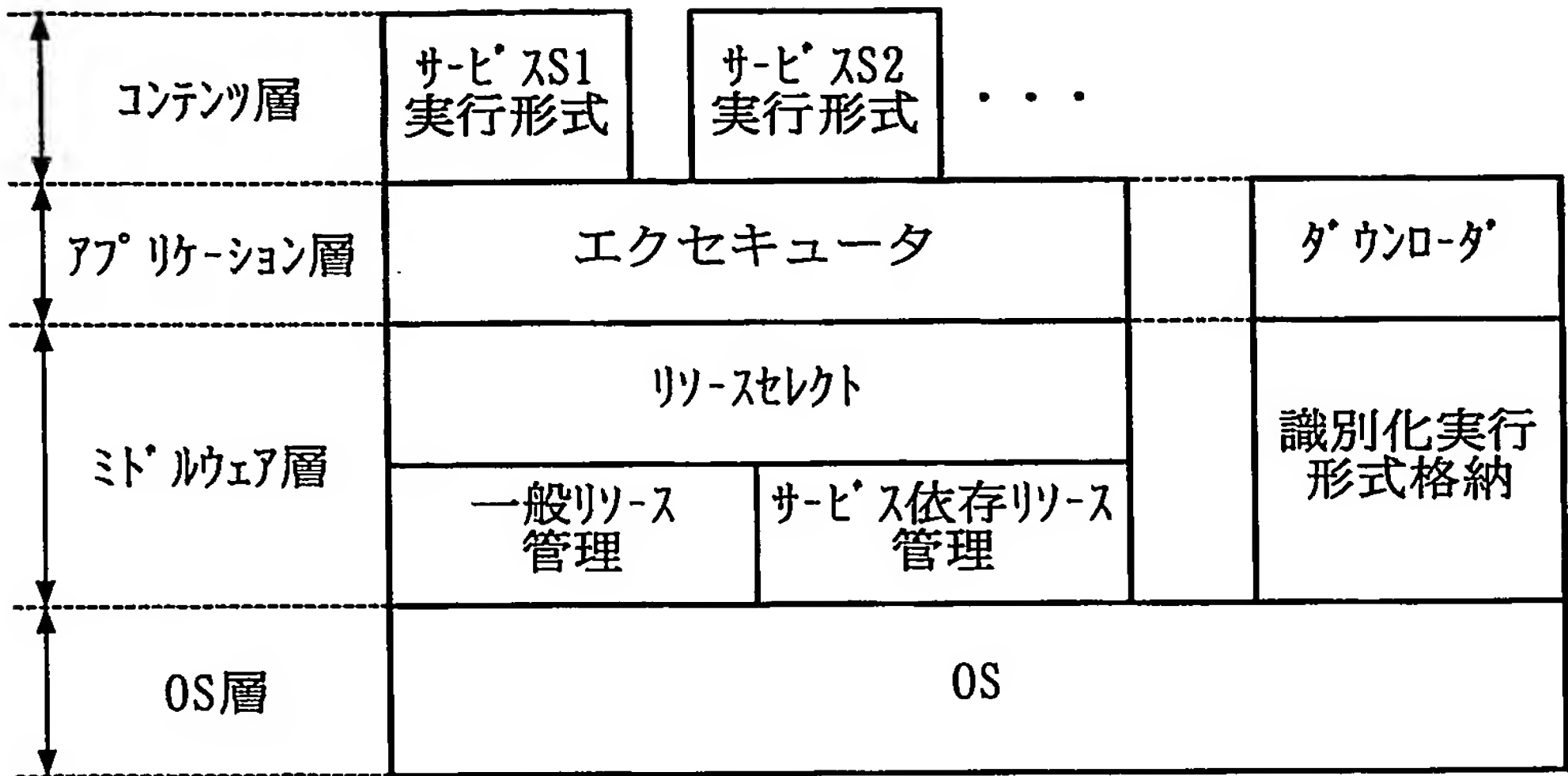


図 7

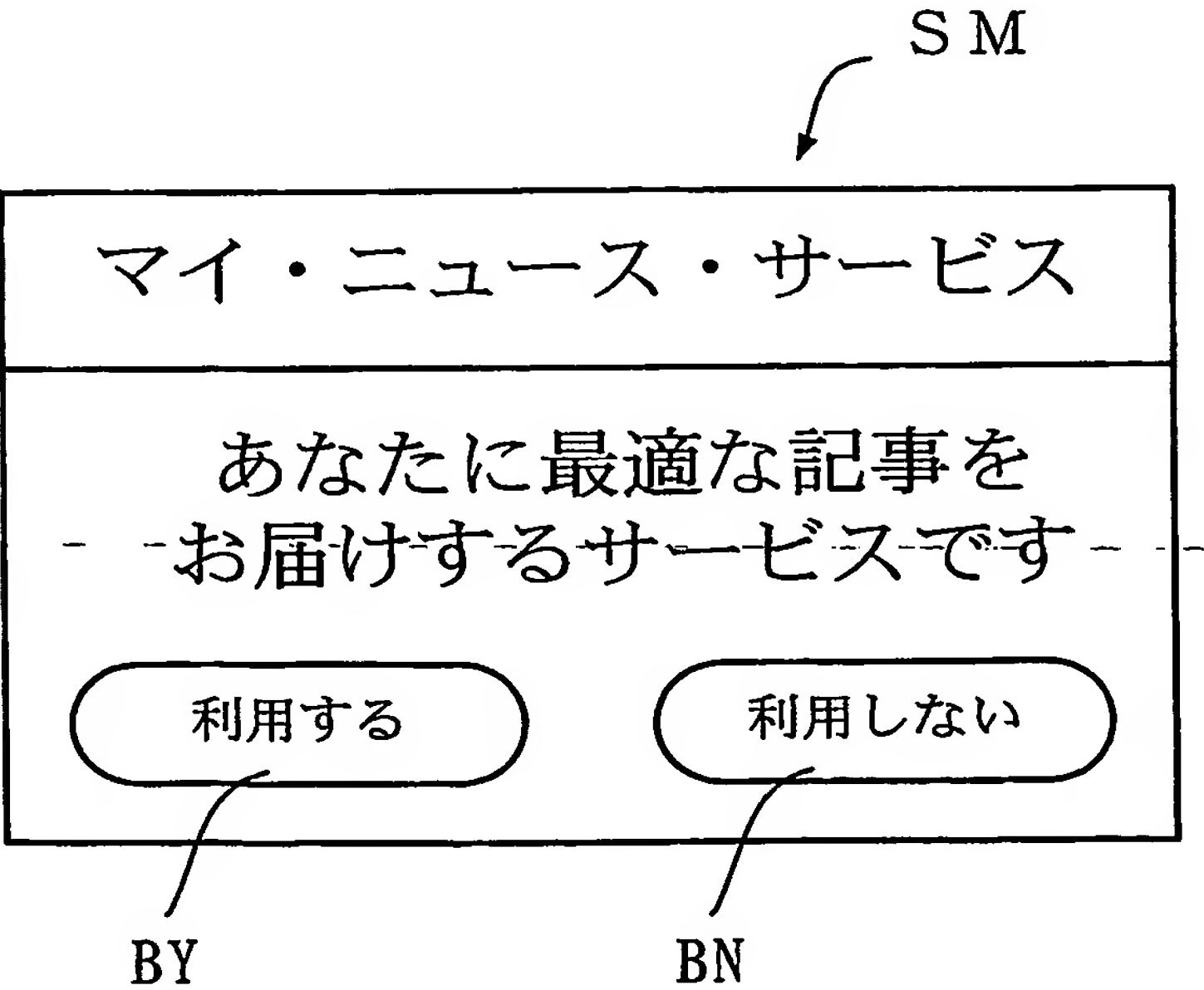


図 8

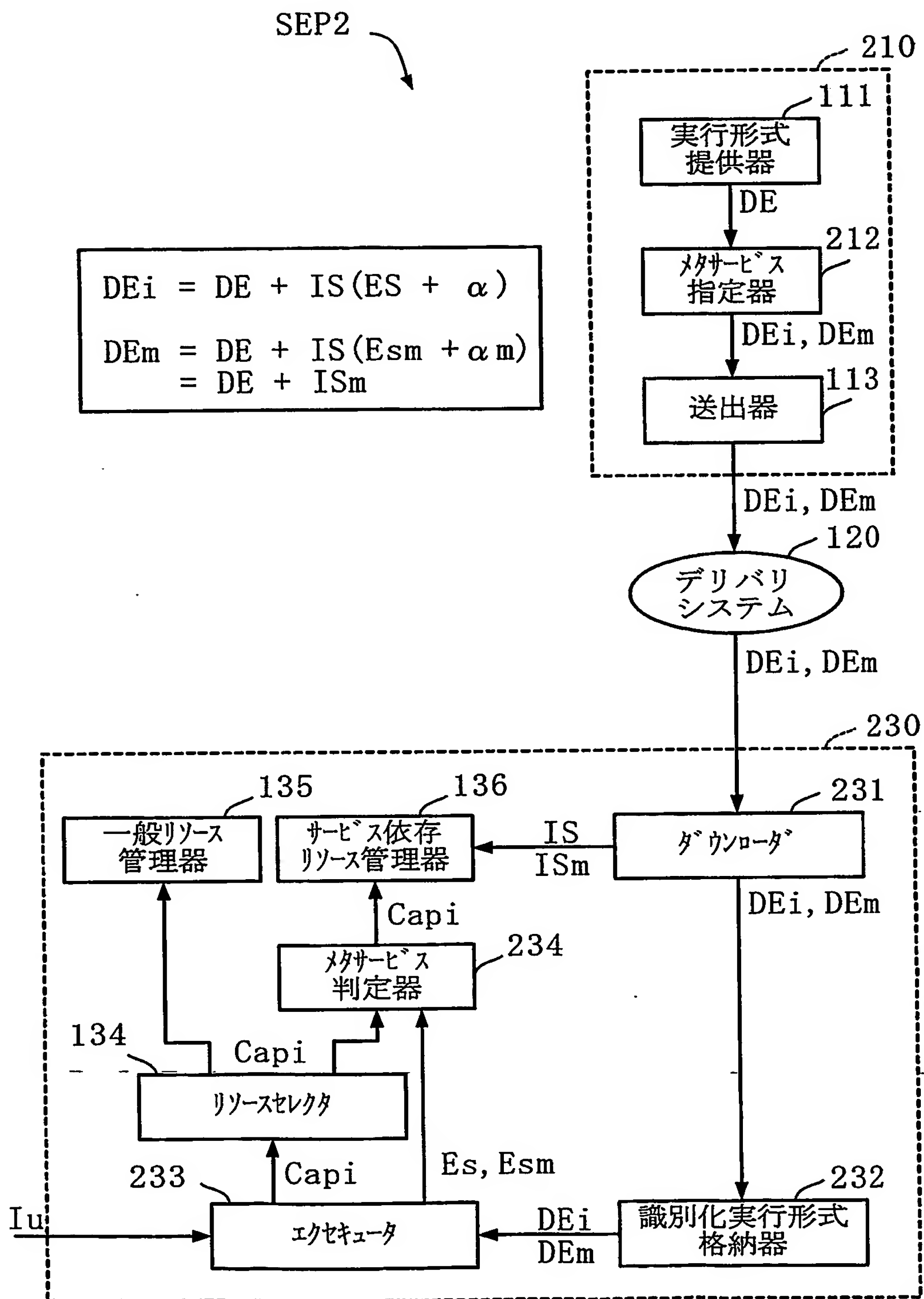


図 9

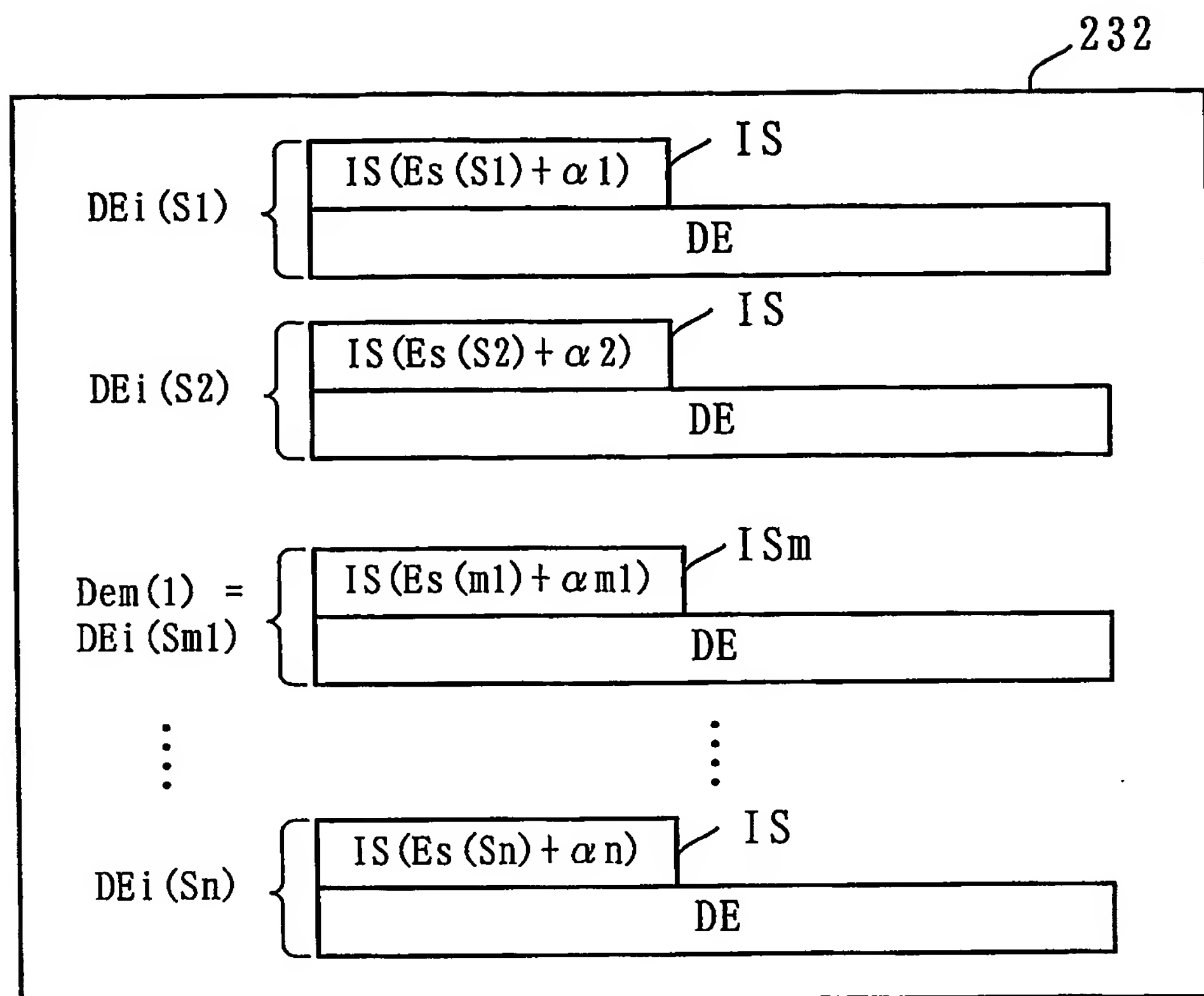


図 10

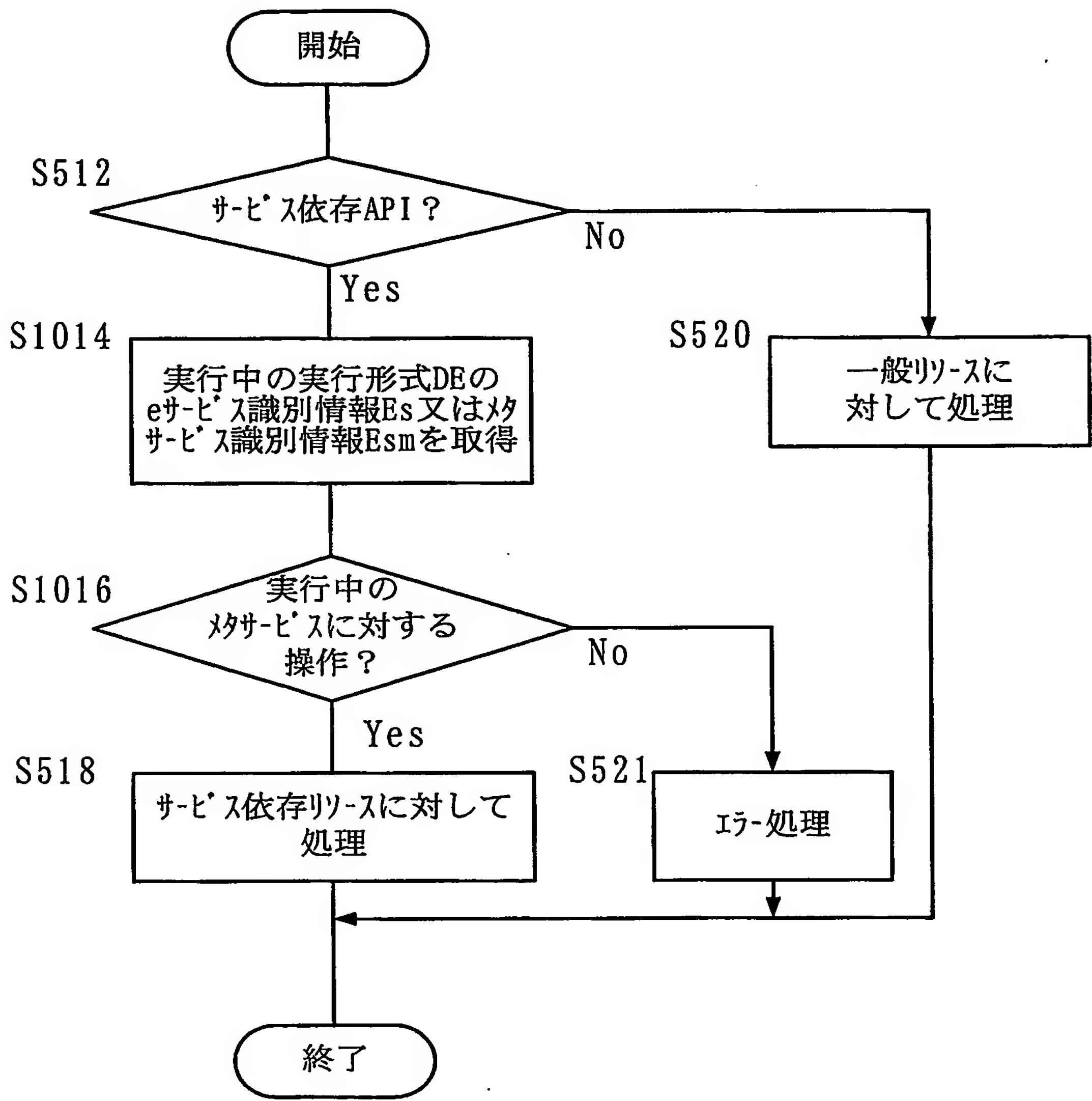




図 11

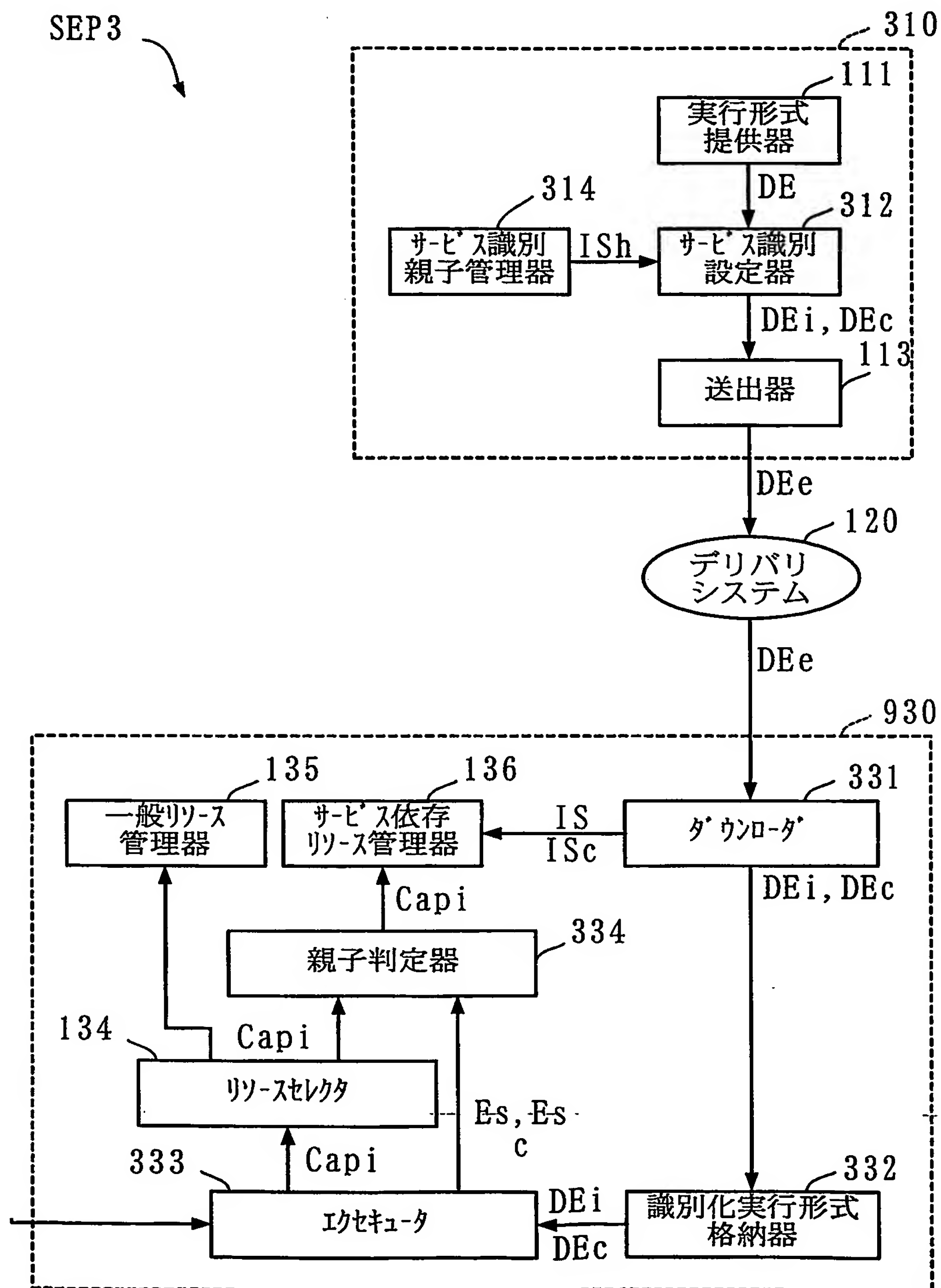


図 1 2

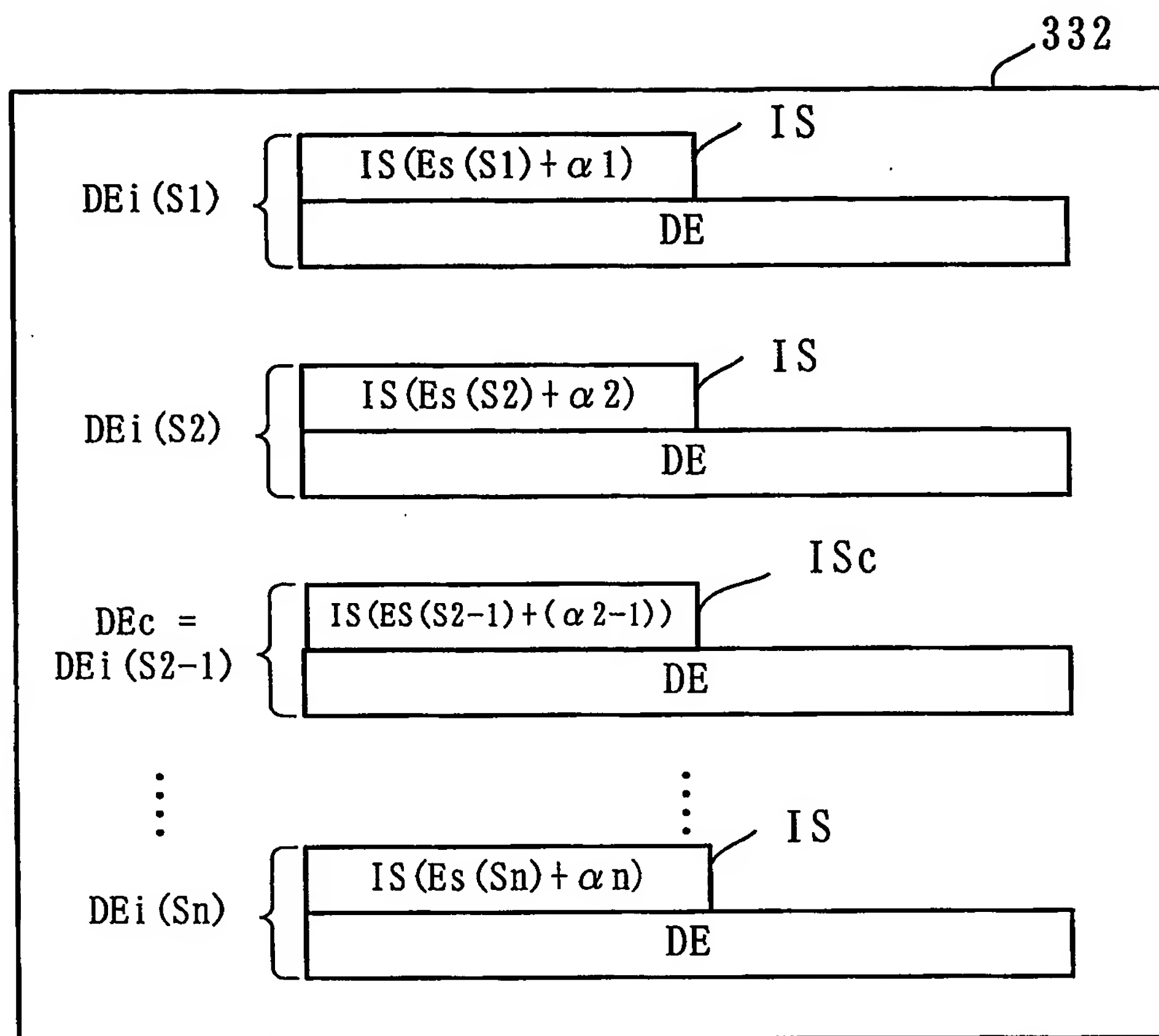


図 13

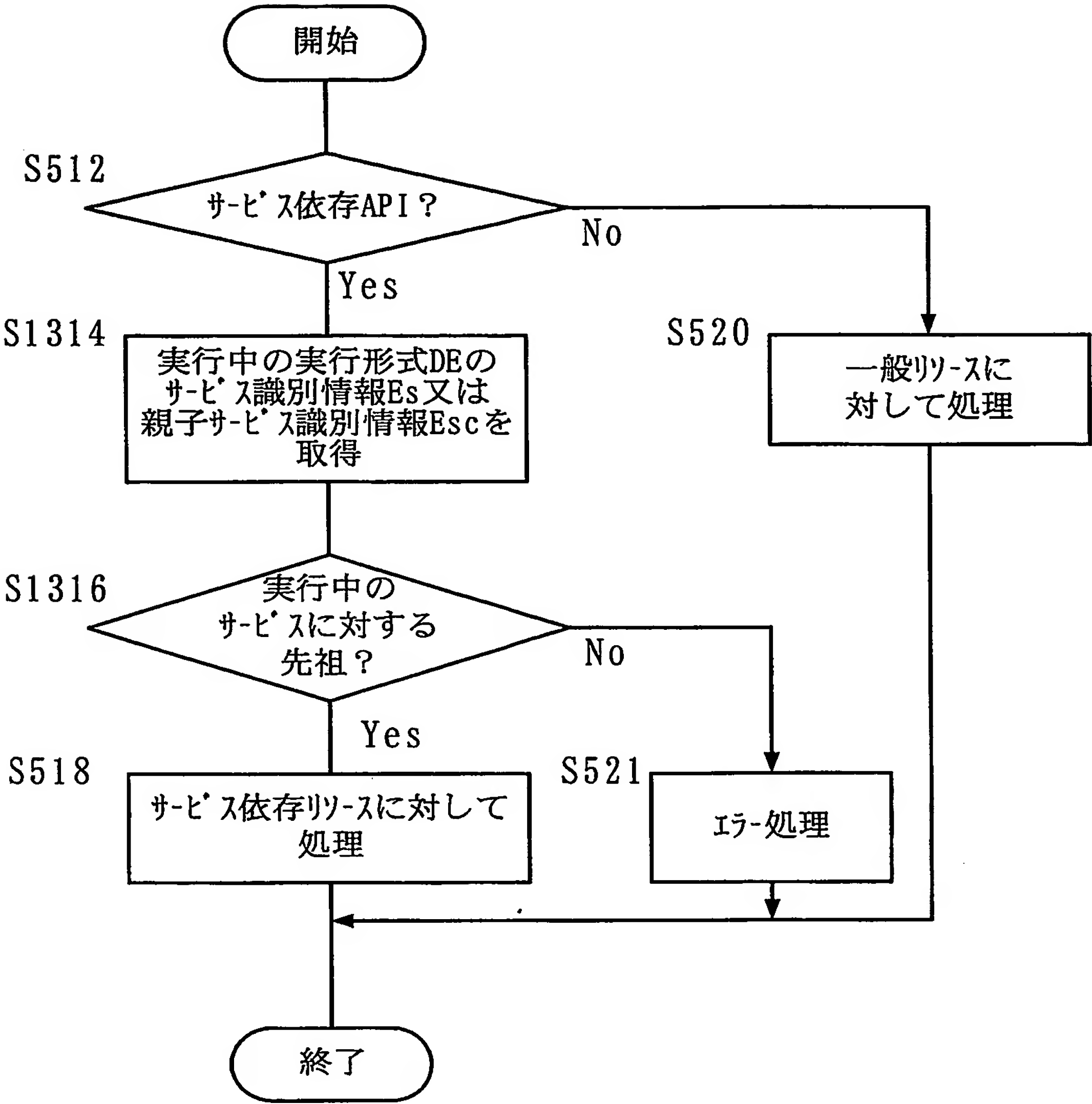


図 14

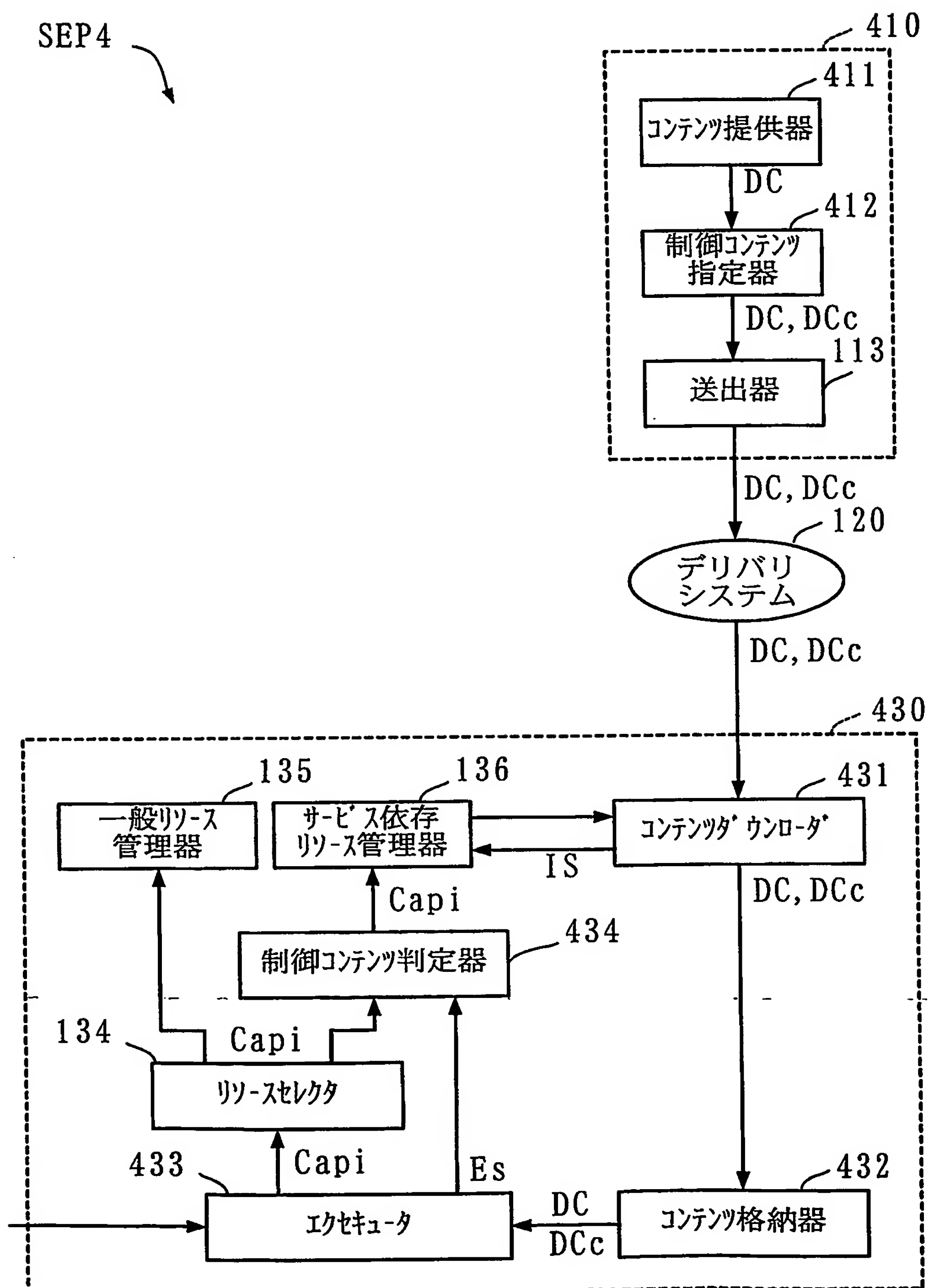


図 1 5

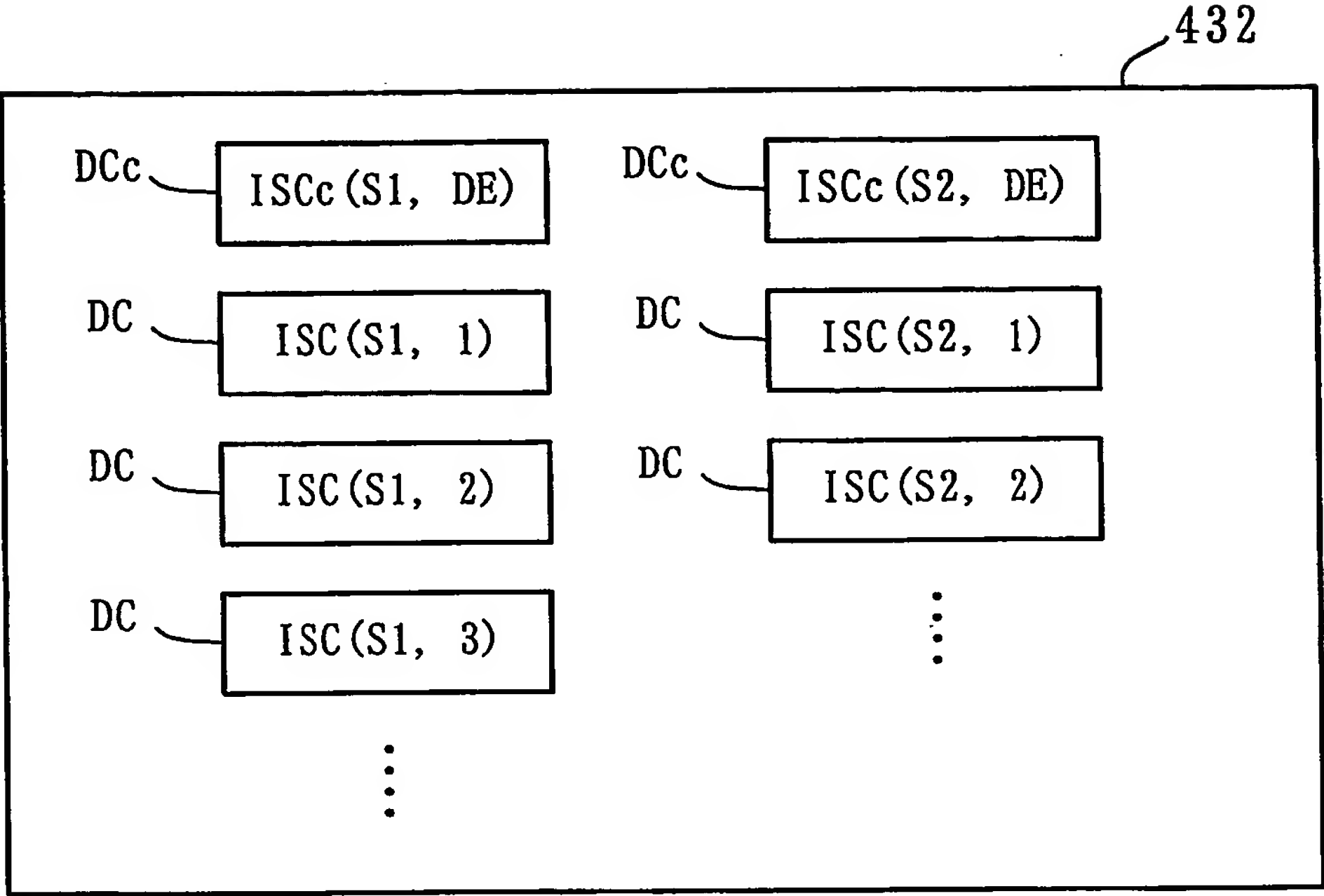


図 1 6

C1

C2

Trs4

	サービス識別 Es	コンテンツ自動 ダウンロード状態
L1 →	S1	ダウンロードする
L2 →	S2	ダウンロードしない
	⋮	⋮

図 17

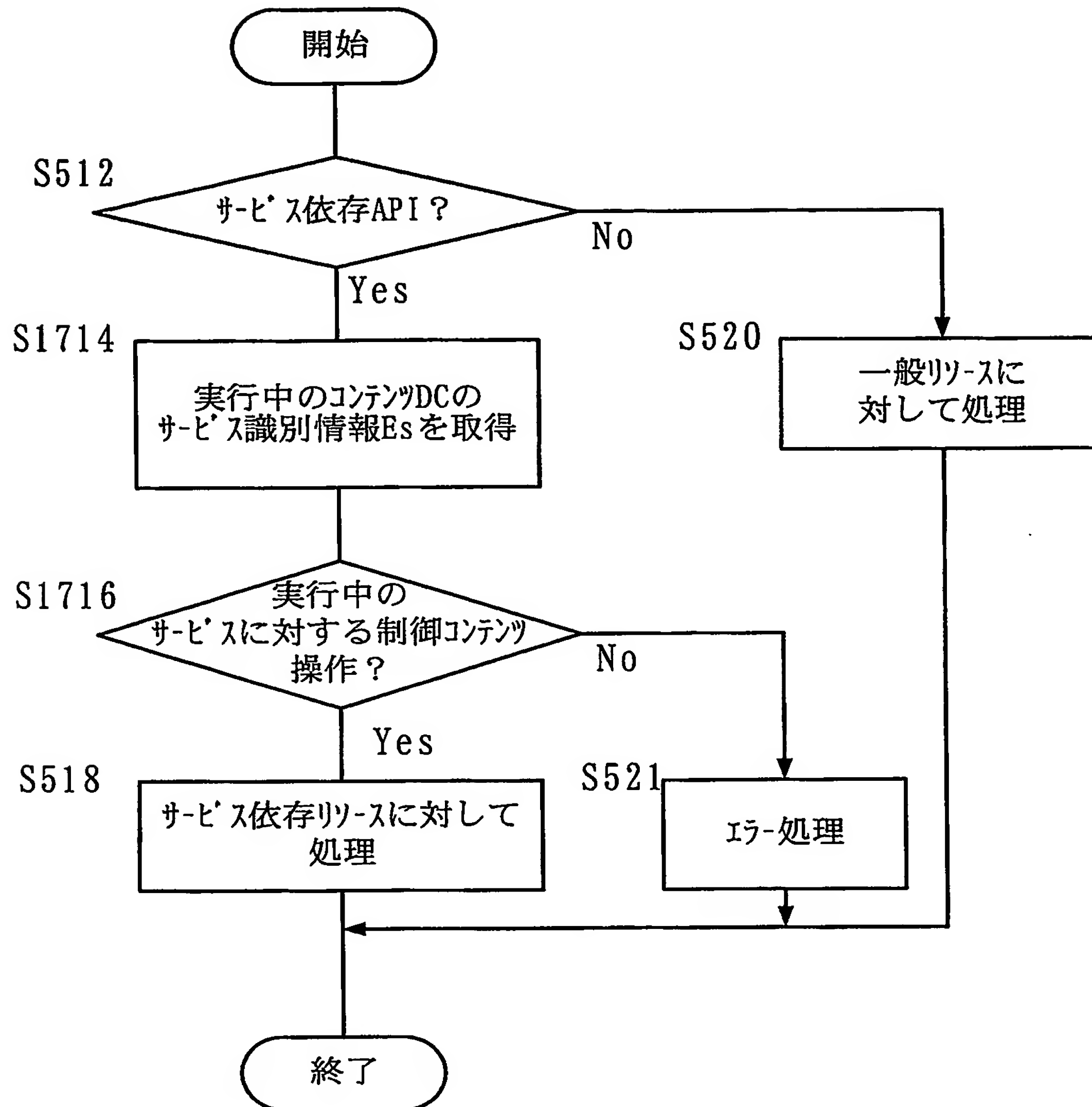




図 18

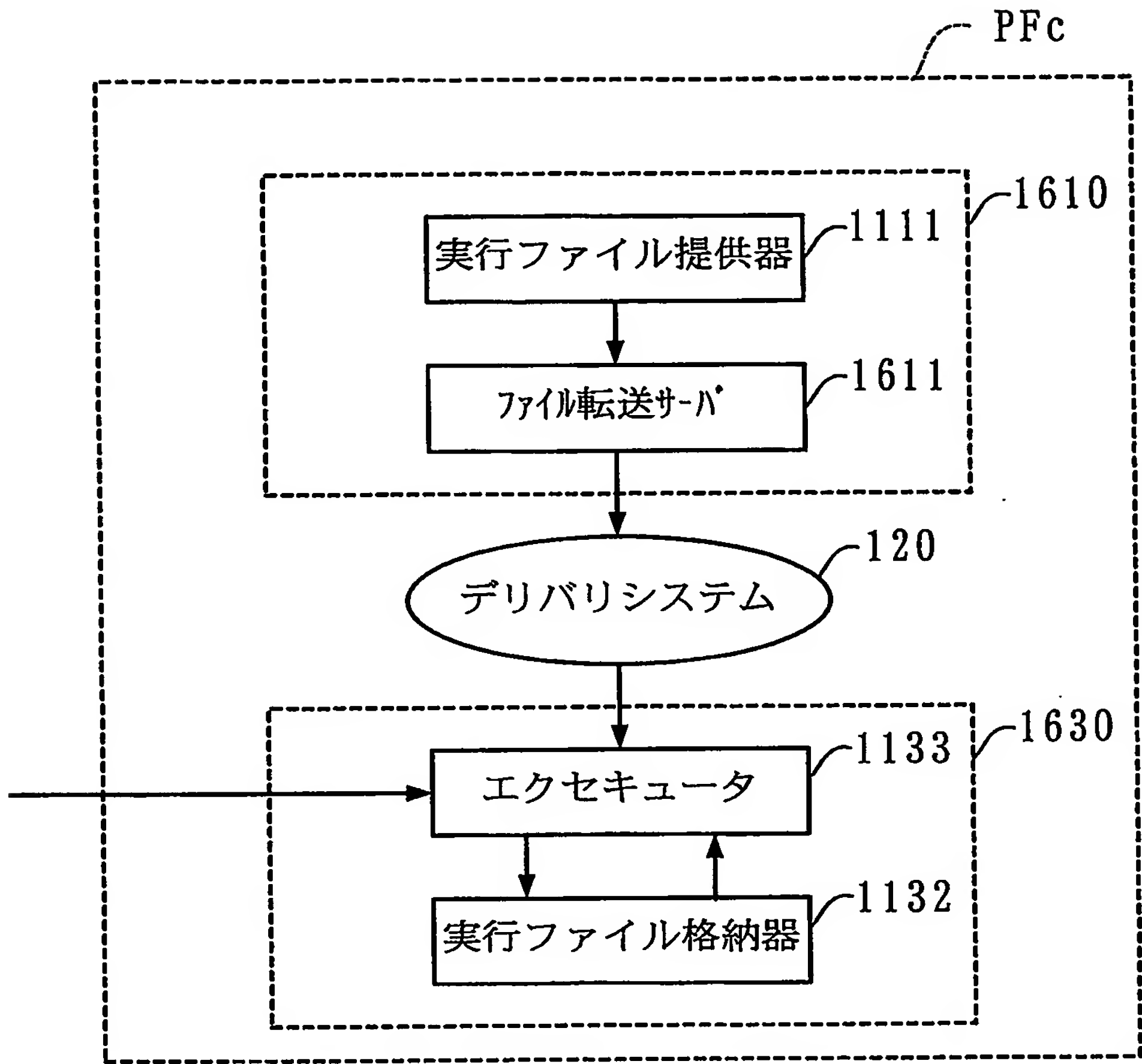


図 1 9

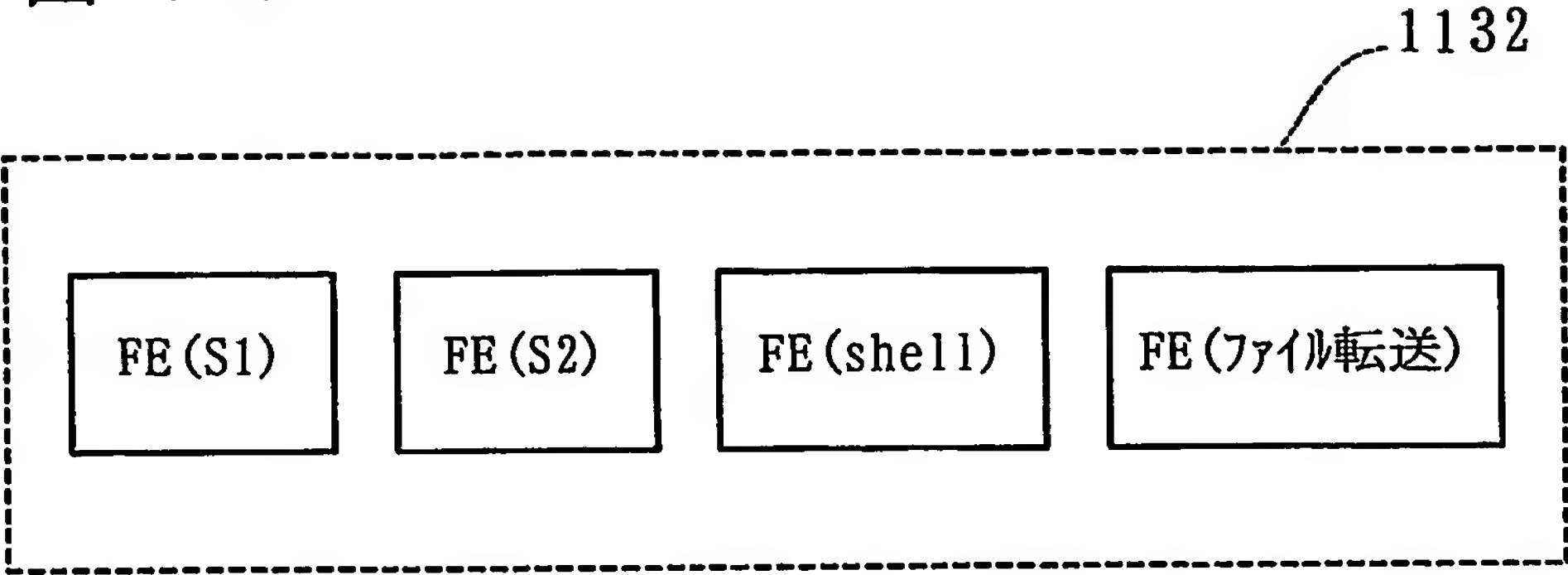
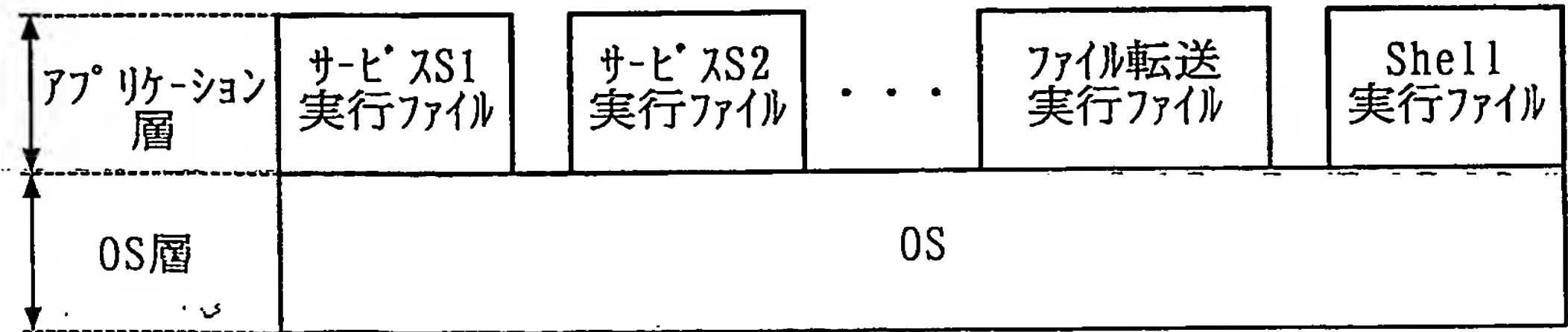


図 2 0



# INTERNATIONAL SEARCH REPORT

International application No.

PCT/JP03/07390

## A. CLASSIFICATION OF SUBJECT MATTER

Int.Cl<sup>7</sup> G06F9/445

According to International Patent Classification (IPC) or to both national classification and IPC

## B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

Int.Cl<sup>7</sup> G06F9/445

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Jitsuyo Shinan Koho	1926-1996	Jitsuyo Shinan Toroku Koho	1996-2003
Kokai Jitsuyo Shinan Koho	1971-2003	Toroku Jitsuyo Shinan Koho	1994-2003

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

## C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y A	JP 2001-5794 A (NEC Corp., Chugoku Nihon Denki Software Kabushiki Kaisha), 12 January, 2001 (12.01.01), Full text; all drawings (Family: none)	1-4 5-12
Y A	JP 2000-172646 A (Toshiba Corp.), 23 June, 2000 (23.06.00), Full text; all drawings (Family: none)	1-4 5-12

☐ Further documents are listed in the continuation of Box C. ☐ See patent family annex.

* Special categories of cited documents:	"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention-
"A" document defining the general state of the art which is not considered to be of particular relevance	"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
"E" earlier document but published on or after the international filing date	"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)	"&" document member of the same patent family
"O" document referring to an oral disclosure, use, exhibition or other means	
"P" document published prior to the international filing date but later than the priority date claimed	

Date of the actual completion of the international search  
16 September, 2003 (16.09.03)

Date of mailing of the international search report  
07 October, 2003 (07.10.03)

Name and mailing address of the ISA/  
Japanese Patent Office

Authorized officer

Facsimile No.

Telephone No.

## A. 発明の属する分野の分類 (国際特許分類 (IPC))

Int Cl<sup>7</sup> G06F9/445

## B. 調査を行った分野

調査を行った最小限資料 (国際特許分類 (IPC))

Int Cl<sup>7</sup> G06F9/445

最小限資料以外の資料で調査を行った分野に含まれるもの

日本国実用新案公報 1926年-1996年

日本国公開実用新案公報 1971年-2003年

日本国実用新案登録公報 1996年-2003年

日本国登録実用新案公報 1994年-2003年

国際調査で使用した電子データベース (データベースの名称、調査に使用した用語)

## C. 関連すると認められる文献

引用文献の カテゴリー*	引用文献名 及び一部の箇所が関連するときは、その関連する箇所の表示	関連する 請求の範囲の番号
Y A	JP 2001-5794 A (日本電気株式会社, 中国日本電気 ソフトウェア株式会社) 2001.01.12, 全文, 全図 (ファミリーなし)	1-4 5-12
Y A	JP 2000-172646 A (株式会社東芝) 2000.06.23, 全文, 全図 (ファミリーなし)	1-4 5-12

☐ C欄の続きにも文献が列举されている。☐ パテントファミリーに関する別紙を参照。

## \* 引用文献のカテゴリー

「A」 特に関連のある文献ではなく、一般的技術水準を示すもの

「E」 国際出願日前の出願または特許であるが、国際出願日以後に公表されたもの

「L」 優先権主張に疑義を提起する文献又は他の文献の発行日若しくは他の特別な理由を確立するために引用する文献 (理由を付す)

「O」 口頭による開示、使用、展示等に言及する文献

「P」 国際出願日前で、かつ優先権の主張の基礎となる出願

の日の後に公表された文献

「T」 国際出願日又は優先日後に公表された文献であって出願と矛盾するものではなく、発明の原理又は理論の理解のために引用するもの

「X」 特に関連のある文献であって、当該文献のみで発明の新規性又は進歩性がないと考えられるもの

「Y」 特に関連のある文献であって、当該文献と他の1以上の文献との、当業者にとって自明である組合せによって進歩性がないと考えられるもの

「&amp;」 同一パテントファミリー文献

国際調査を完了した日

16.09.03

国際調査報告の発送日 07.10.03

国際調査機関の名称及びあて先

日本国特許庁 (ISA/JP)

郵便番号100-8915

東京都千代田区霞が関三丁目4番3号

特許庁審査官 (権限のある職員)

漆原 孝治

5B

9366

電話番号 03-3581-1101 内線 3546